IBM WebSphere Commerce
for AIX

IBM

Migration Guide

Version 54

IBM WebSphere Commerce
for AIX

IBM

Migration Guide

Version 54

> **Note:**
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page 221.

Contents

About this book

> **Important**
>
> Before proceeding with the migration steps described in this Guide, download the latest WebSphere Commerce 5.4 FixPak from the WebSphere Commerce Downloads Web site.
>
> - For Business Edition:
>
>   `http://www.ibm.com/software/webservers/commerce/wc_be/downloads.html`
>
> - For Professional Edition:
>
>   `http://www.ibm.com/software/webservers/commerce/wc_pe/downloads.html`
>
> You will need to apply this WebSphere Commerce 5.4 FixPak *after* you upgrade your software from WebSphere Commerce Suite 5.1 to WebSphere Commerce 5.4, but *prior* to migrating your assets, such as the commerce database, instance, configuration, and code. The FixPak contains corrections that will minimize any problems you may encounter in the migration process.
>
> Refer to the installation instructions that accompany the FixPak for details on how to apply it.
>
> Refer to the *WebSphere Commerce 5.4 Installation Guide* for additional installation considerations when installing and configuring the WebSphere Commerce 5.4 software stack. To install additional software packaged with WebSphere Commerce 5.4, see the *WebSphere Commerce 5.4 Additional Software Guide*.

This document describes the steps to migrate WebSphere Commerce Suite 5.1 to WebSphere Commerce 5.4 on AIX®Linux. If you wish to migrate WebSphere Commerce Studio 5.1 to WebSphere Commerce Studio 5.4, see the *WebSphere Commerce Studio 5.4 Migration Guide*.

In this document, any reference to WebSphere Commerce 5.4 or WebSphere Commerce mean the current release, WebSphere Commerce 5.4. References to WebSphere Commerce Suite or Commerce Suite mean the previous release, Commerce Suite 5.1 along with Commerce Suite 5.1.1, Commerce Suite 5.1.1.1, and Commerce Suite 5.1.1.2.

Summary of changes

The most recent version of this document is avilable as a PDF file from the Technical Library page of the WebSphere Commerce Web site:

- Business Edition:

  `http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html`

- Professional Edition:

  `http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html`

Updates from the last version of this book are identified by revision characters contained in the margin. This book uses the following conventions for revision characters:

- The "+" (plus) character identifies updates that have been made in the current version of this book.
- The "|" (split vertical bar) character identifies updates that have been made in the previous versions of this book.

The following table shows the main changes that have been made to this book.

Change	Chapters or pages affected
Updated product.xml example	"Updating product.xml" on page 66
Added information on the "Migration Bundle"	3 in Chapter 1, "Premigration," on page 3 and at the end of "Upgrading to WebSphere Commerce 5.4" on page 58
Added information in Troubleshooting section if a 404 error is received when accessing WebSphere Commerce tools	At the beginning of Appendix H, "Troubleshooting," on page 211
Added information on the iiDB.getDate() method	"Change to iiDB.getDate() method in InterestItemDisplay.jsp" on page 114
MigrateEncryptedInfo script updates due to Migration Bundle	"Migrating your security configuration" on page 104
Updates to output from premigration checker	"Optional database premigration items" on page 77

To learn about last-minute changes to the product, see the updated product README file, also available from the WebSphere Commerce Web site.

If you are installing WebSphere Commerce 5.4 on a machine that does *not* have earlier versions of Commerce Suite and supported products installed, see the *WebSphere Commerce 5.4 Installation Guide*.

What's new in WebSphere Commerce 5.4

Important enhancements and new features have been added to WebSphere Commerce 5.4 from the previous release of WebSphere Commerce. For detailed information on the enhancements and new features available with this release, see the *What's New In IBM WebSphere Commerce, 5.4* document.

Enhancements have been made in the following areas:
- Access control
- Advanced user, member, and organization management
- Billing, invoicing, and credit management
- Business relationship management
- Buy-side purchasing
- Catalog Manager
- Catalog subsystem
- Collaboration
- Commerce Accelerator enhancements
- Commerce Models
- Configuration Manager
- Contract based commerce
- Coupons
- Inventory management

- Loader package
- Marketing subsystem
- Message extensions from Commerce Integrator
- Order management enhancements
- Organization Administration Console
- WebSphere Payment Manager
- Product Management
- Request for Quotation (RFQ)
- Returns and refunds including RMA functionality
- Search Enhancements
- Security Enhancements
- Store Services
- WebSphere Application Server 4.0.2
- WebSphere® Commerce Analyzer Enhancements
- XML over HTTP

After you have migrated your Commerce Suite 5.1 system to WebSphere Commerce 5.4 as described in this Migration Guide, check the *What's New* document and the WebSphere Commerce 5.4 online help for information on how to use these new features in your migrated system.

Who should perform the migration?

Since the migration tasks are quite technical, the bulk of these tasks should be performed by a System Administrator. The following is a summary of the various users and their expected role in the migration process:

System Administrator
Knowledge and experience:
- Familiar with programming (for example, Java™, JavaServer Pages, and so on)
- Familiar with database administration (DB2® or Oracle)
- Web master
- Knowledgeable on System Architecture

Tasks:
- Installs, configures, and maintains WebSphere Commerce.
- Manages the database.
- Manages the Web server.
- Controls access.
- Manages data updates through mass import or other mechanisms.

Expectations of the migration process:

The migration programs and procedures should allow the current **system assets** *to be migrated with minimum downtime.*

Store Developer
Knowledge and experience:
- Familiar with programming.
- Familiar with multimedia tools

Tasks:
- Creates and customizes the store.
- Sets up and customizes payment, shipping, and tax support.

Expectations of the migration process:

The migration programs and procedures should allow the current **store assets** *to be migrated with minimum downtime.*

Store Administrator

Knowledge and experience:
- Familiar with business-procedures.
- Web-literate.

Tasks:
- Manages orders
- Processes payments
- Assists shoppers
- Maintains online store
- Modifies online store

Expectations of the migration process:

Information acquired online such as orders and shoppers will be available and usable after migration.

Catalog Administrator

Knowledge and experience:
- Is a product expert.
- Is Web-literate and computer-literate.
- Is familiar with multimedia tools.

Tasks:
- Creates store catalogs
- Creates and manages products and categories
- Creates and manages pricing schemes
- Creates and manages reports

Expectations of the migration process:

Information using a previous version of WebSphere Commerce Suite will not have to be recreated. Tools will be adaptable to extensions to the catalog.

Supported migration paths

> **Note:**
> This Guide describes the migration steps to migrate to WebSphere Commerce 5.4 Business Edition or Professional Edition on the AIX platform.

The following migration paths for all national language versions are supported:

- Commerce Suite 5.1 Pro Edition to WebSphere Commerce 5.4 Professional Edition
- Commerce Suite 5.1 Pro Edition to WebSphere Commerce 5.4 Business Edition
- Commerce Suite 5.1.1.1 to WebSphere Commerce 5.4
- Commerce Suite 5.1.1.2 to WebSphere Commerce 5.4

Attention: This Migration Guide has been tested only for the above migration paths. This Guide does not address migration to WebSphere Commerce 5.4 from any version of WebSphere Commerce Suite later than the above versions, that is, any version of Commerce Suite that has been upgraded by applying any potential future FixPaks, eFixes, or other enhancements.

For any information on migration issues arising from the application of any future FixPaks, eFixes, or other enhancements, see the WebSphere Commerce Support Web page at:

http://www.ibm.com/software/webservers/commerce/support.html

Migrating from earlier versions

This guide describes the migration process for the above supported migration paths. Direct migration from earlier versions of Net.Commerce™ or Commerce Suite is not supported.

To migrate from earlier versions of Net.Commerce or Commerce Suite to WebSphere Commerce 5.4, you must first migrate your existing system to the WebSphere Commerce 5.1 level, and then use this document to migrate to WebSphere Commerce 5.4.

- To transition and migrate to Commerce Suite 5.1 from earlier versions of Net.Commerce, see the *WebSphere Commerce Suite 5.1 Migration and Transition Guide*.

Migration paths by operating systems

WebSphere Commerce 5.4 supports the migration of Commerce Suite on the same operating system:

- AIX to AIX

WebSphere Commerce 5.4 does not support migration between dissimilar operating systems, for example, Commerce Suite 5.1 on Windows NT® to WebSphere Commerce 5.4 on AIX.

Non-supported migration paths

WebSphere Commerce 5.4 does not support the following migration path:

- Migration from one database vendor product to another (for example, DB2 to Oracle®).

Other considerations

- Merging multiple national language version databases into one database that supports multiple languages. Consult IBM® Global Services for assistance if you wish to do this.
- When you install a WebSphere Commerce 5.4 system on a machine that has WebSphere Commerce Suite 5.1 installed, the directory for WebSphere

Commerce Suite 5.1 will be preserved as you need to copy over the old store assets to the new directory tree. WebSphere Commerce Suite 5.1 will cease to function.

- If you have customized code or commands written in Java or Enterprise JavaBeans™ objects in WebSphere Commerce 5.1 or WebSphere Commerce Business Edition 5.1, you need to redeploy your them to the level required for WebSphere Commerce 5.4. See the "Transitioning your customized code" section of the *WebSphere Commerce Studio 5.4 Migration Guide*. This transition is required due to the move to WebSphere Application Server 4.0.2 from IBM WebSphere Application Server 3.5.

Conventions used in this book

This book uses the following highlighting conventions:

- **Boldface type** indicates commands or graphical user interface (GUI) controls such as names of fields, icons, or menu choices.
- `Monospace type` indicates examples of text you enter exactly as shown, file names, and directory paths and names.
- *Italic type* is used to emphasize words. Italics also indicate names for which you must substitute the appropriate values for your system. When you see any of the following names, substitute your system value as described:

host_name
> The fully qualified host name of your WebSphere Commerce server (for example, `ibm.com` is fully qualified).

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

Note: References to "Commerce Suite 5.1" in this document should be understood to mean either Commerce Suite 5.1, Commerce Suite 5.1.1.1, or Commerce Suite 5.1.1.2, unless specifically indicated.

Part 1. Required migration steps

The chapters in this part of the Migration Guide describe the mandatory tasks that are required to migrate to WebSphere Commerce Business Edition 5.1 from Commerce Suite 5.1. They include:

- Chapter 1, "Premigration," on page 3
- Chapter 2, "Backing up Commerce Suite 5.1," on page 27
- Chapter 3, "Upgrading your software," on page 35
- Chapter 4, "Migrating your database," on page 69
- Chapter 5, "Post data migration actions," on page 99

In addition, Chapter 6, "Additional post data migration actions," on page 119 describes optional post migration actions that you may wish to take depending on your requirements.

Chapter 1. Premigration

The following sections describe specific steps you must complete while you still have an operational Commerce Suite 5.1 system. The sections also describe some considerations you must make prior to beginning your migration to WebSphere Commerce 5.4.

> **Important**
>
> If you have customized code or commands written in Java or Enterprise JavaBeans in WebSphere Commerce 5.1 or WebSphere Commerce Business Edition 5.1, you need to redeploy your them to the level required for WebSphere Commerce 5.4. See the "Transitioning your customized code" section of the *WebSphere Commerce Studio 5.4 Migration Guide*. This transition is required due to the move to WebSphere Application Server 4.0.2 from IBM WebSphere Application Server 3.5.
>
> If you have customized some EJBs, confirm the JNDI names are the same as they are called in customized commands after migration.
>
> If any of your JSPs or customized code previously used the `com.ibm.util` package from IBM WebSphere Application Server 3.5.x, you will have to rewrite your code or JSP by using the equivalent class from the IBM SDK for Java that is packaged with WebSphere Application Server 4.0.2. The `com.ibm.util` package has been removed from the current version of WebSphere Application Server.
>
> After you install WebSphere Commerce 5.4, you need to install a migration bundle that is available from the WebSphere Commerce Support site (`http://www.ibm.com/software/commerce/support/`). Click on the link for the edition you are using and see the section **Software downloads**. Download the migration bundle, APAR IY46716 and follow the instructions that are provided in the bundle to install it. You should not proceed with the migration steps in this Guide (such as migrating your instance or database) without installing this migration bundle.

Premigration actions

Before proceeding with the migration steps contained in the rest of this guide, ensure that you take the following actions while your system is operational at the Commerce Suite 5.1 level.

- In order to easily migrate your Commerce Suite 5.1 store archives, you need to publish them while you are at the Commerce Suite 5.1 level, prior to migrating to WebSphere Commerce 5.4. See the section "Publish a store archive" in the Commerce Suite 5.1 online help for the detailed steps to publish a store while at the Commerce Suite 5.1 level.
- If you are going to migrate your inventory representation to use the new ATP inventory representation, you must close all existing auctions in your Commerce Suite 5.1 system before migrating to WebSphere Commerce 5.4. See the section "Close bidding for an auction" in the Commerce Suite 5.1 online help for information on closing your auctions.

- If you have customized any property files that are supplied with Commerce Suite 5.1, make a copy of them to a directory that will be accessible to you after you complete the migration, say on a different drive than where Commerce Suite 5.1 is installed. For example, back up the UserRegistration_en_US.properties file in the /usr/lpp/CommerceSuite/properties directory.
- The lengths of the following columns have changed from Commerce Suite 5.1 to WebSphere Commerce 5.4. In order to ensure that you do not lose any data contained in these columns during the migration process, make sure that your data in these columns in Commerce Suite 5.1 do not exceed the new length for the column. For example, ensure that MBRGRP.DESCRIPTION is not greater than 512 characters.

Table.Column	Commerce Suite 5.1	WebSphere Commerce 5.4
MBRGRP.DESCRIPTION	LONG VARCHAR	VARCHAR(512)
ORGENTITY.DESCRIPTION	LONG VARCHAR	VARCHAR(512)
CONTRACT.NAME	VARCHAR(254)	VARCHAR(200)

- **If you are using Oracle**, and if you have extended any Commerce Suite 5.1 tables, there is a possibility that the migration script will experience problems if you have defined a column as LONG. Check your Commerce Suite 5.1 tables, for example, the CATENTDESC table.

  If you have not added a column defined as LONG, then there is no need to take any action. If you have added any columns defined as LONG to these tables, then you need to regenerate the indexfile.sql by using the instructions below:

  1. Export the table with the columns defined as LONG to a file (for example, export table CATENTDESC).
  2. Drop or delete the table.
  3. Import the table using the indexfile option. This will create a file that will create the table.
  4. Edit the file created in the above step, and change the column definitions to the one that you want. Refer to the indexfile.sql sample file provided in the /usr/WebSphere/CommerceServer/schema/oracle/migration directory.
  5. Run the file you edited in the above step to create the new table.
  6. Re-import the data by running import with ignore=y.
  7. Start the data migration process once again.

  Also, see the related troubleshooting item in Appendix H, "Troubleshooting," on page 211 for more information.

- Drop any of the constraints that are connected to Commerce Suite 5.1 tables before migration, and then restore the constraints after migration to WebSphere Commerce 5.4. If you have customized tables that contain foreign key links to the shipped Commerce Suite 5.1 tables, the data migration script may fail when it attempts to drop the referential integrity constraints (such as foreign keys, primary keys, and indexes) during data migration. You can use the sample SQL statements in "Dropping custom constraints" on page 80 to drop these constraints. After you have migrated your data, using the supplied data migration script, you need to add back the constraints as described in "Restoring custom constraints" on page 85.

  If you have modified any of the Commerce Suite 5.1 referential integrity constraints, that is, you have added an extra column to an already existing index or foreign key relationship, the column will be deleted as part of the data migration process.

You should refer to the documentation provided by your database vendor for more information on referential constraints.

- If you have customized database tables in Commerce Suite 5.1 and wish to use the Staging server in WebSphere Commerce 5.4, you need to run the Stage Copy utility command (`stagingcopy`) prior to the data migration to keep the product database and staging database consistent. See the section ″Stage Copy utility command″ in the Commerce Suite 5.1 online help for more information.

- The unique index in the `CONTRACT` table is now constructed based on the following columns:
  - `NAME`
  - `MEMBER_ID`
  - `MAJORVERSION`
  - `MINORVERSION`
  - `ORIGIN`

  If the `MEMBER_ID`, `MAJORVERSION`, and `MINORVERSION` values are the same across two or more rows, you need to make sure that the `NAME` column in the `CONTRACT` table is different across these rows to ensure uniqueness. Otherwise, you may encounter problems when migrating your data from WebSphere Commerce 5.1.

Premigration considerations

The following sections highlight some key items for WebSphere Commerce 5.4 that you need to consider prior to beginning the migration process.

> **Important**
>
> The database migration using the supplied database migration script is a key part of the migration process from Commerce Suite 5.1 to WebSphere Commerce 5.4. Before you begin migrating your Commerce Suite 5.1 database, it is highly recommended that you review the description of the database migration script and understand the settings it uses while migrating your database.

Migrating to ATP inventory

Previous versions of WebSphere Commerce provided task command interfaces to check and update available inventory levels. The default task command implementations used the `INVENTORY` table to record available inventory levels. This previous level of capability is referred to as *Compatibility Mode Inventory*.

Table 1. Compatibility Mode Inventory

Compatibility Mode Inventory Task Command Interface	Description	Called By
ResolveFulfillmentCenterCmd	Determines the FulfillmentCenter for an OrderItem.	OrderItemAdd, OrderItemUpdate, OrderPrepare
CheckInventoryCmd	Checks to see if there is sufficient available inventory for an item.	ResolveFulfillmentCenterCmd
UpdateInventoryCmd	Decreases the available inventory for an item.	OrderProcessCmd, PaySynchronizePM
ReverseUpdateInventory	Increases the available inventory for an item.	Order Management user interface when canceling an Order.

WebSphere Commerce 5.4 enhances this capability with new task command interfaces to check, allocate, or back order available or expected inventory items. The new default task command implementations use information in the RECEIPT, RADETAIL, and other related tables. When items are checked or back ordered, estimated availability times are obtained. This enhanced capability is referred to as *Available to Promise* (ATP) Inventory. Allocations and back orders can expire if payment is not initiated on time.

Table 2. Available to Promise (ATP) Inventory

ATP Inventory Task Command Interface	Description	Called By
AllocateInventoryCmd	Checks, allocates, or back orders available or expected inventory. Determines FulfillmentCenters and estimated availability times. Can also be used to reverse allocations or back orders.	OrderItemAdd, OrderItemUpdate, OrderPrepare, OrderProcess, ProcessBackOrders
GetEligibleFulfillmentCentersCmd	Determines a prioritized list of FulfillmentCenters.	AllocateInventoryCmd
CheckInventoryAvailabilityCmd	Obtains estimated availability times.	AllocateInventoryCmd
AllocateExistingInventoryCmd	Allocates available inventory.	AllocateInventoryCmd
DeallocateExistingInventoryCmd	Reverses an allocation.	AllocateInventoryCmd, ReleaseExpiredAllocations
AllocateExpectedInventoryCmd	Creates a back order.	AllocateInventoryCmd
DeallocateExpectedInventoryCmd	Reverses a back order.	AllocateInventoryCmd, ReleaseExpiredAllocations

The OrderItemAdd, OrderItemUpdate, and OrderPrepare commands have new parameters that allow the caller to check, allocate, or back order inventory, or reverse allocations or back orders. OrderProcess always allocates, or, if unable to allocate, back orders inventory for any OrderItems not yet allocated nor back ordered.

Table 3. ATP Inventory Commands

ATP Inventory Enabled Command	Enhanced ATP Capabilities	Default Action
OrderItemAdd, OrderItemUpdate	Check, allocate, back order, reverse.	Check.
OrderPrepare	Check, allocate, back order, reverse.	Allocate or back order.
OrderProcess	Allocate or back order.	Not applicable.

A new column in the STORE table, named ALLOCATIONGOODFOR, is used to enable the ATP Inventory capability for a store. When the value in this column is zero, Compatibility Mode Inventory is enabled. When the value is greater than zero, ATP Inventory is enabled. The value specifies the number of seconds after which allocations and back orders expire, if payment is not yet initiated. When you migrate your database from Commerce Suite 5.1 to WebSphere Commerce 5.4, you can choose to migrate to ATP Inventory. If you do so, then information in the INVENTORY table will be moved to the RECEIPT table, and STORE.ALLOCATIONGOODFOR will be set to its default value (43, 200 seconds or 12 hours).

Catalog entries in the WebSphere Commerce Suite 5.1 INVENTORY table that are either products, DynamickitBean, or bundles, will not be moved to the WebSphere Commerce 5.4 RECEIPT table during the database migration as WebSphere Commerce 5.4 inventory function does not support them. Only catalog entries that

are packages or items in the WebSphere Commerce Suite 5.1 `INVENTORY` table will be moved to the WebSphere Commerce 5.4 `RECEIPT` table during the database migration. The new ATP function supports only these types of catalog entries.

If you choose not to migrate to ATP Inventory at this time, you can do so at a later time using the `migrateATP` script. This script is described in Appendix D, "Converting to ATP inventory at a later time," on page 167.

Orders and order items

There are two premigration considerations you need to make with respect to your Commerce Suite 5.1 orders or order items:

- Orders or order items are in the `M` state (that is, the shopper has initiated payment and the inventory update is successful, but the order or order item has not been authorized).

  In this case, you should either complete, or delete, or cancel the orders or order items as appropriate prior to migration. Usually, orders or order items in this state are simply waiting authorization to proceed, and are only in the `M` state over a short time period. They will however, remain in this state if authorization either fails or gets rejected. If you migrate these orders and order items while they are in the `M` state, WebSphere Commerce 5.4 will run the `PMClean` command as a scheduled job, to clean up these orders and order items.

- Orders or order items are in the `C` state (that is, payment has been authorized).

  If the items are truly complete and have been shipped, you should move the order or order item to a final `S` state (that is, the order item has been shipped). This will prevent the order items from being allocated again in WebSphere Commerce 5.4.

  During the database migration process, a `ctos.sql` script gets generated by the database migration script if you specify the ATP option. The `ctos.sql` script changes the status of any orders or order items with a status of `C`, from `C` to `S`. You should run the `ctos.sql` script in the following cases:

  - If you run the database migration script using the `ATP` option:

    You need to run the `ctos.sql` script prior to restarting the Web server and the `WebSphere Commerce Server — instance_name` application server after migration.

  - If you run the database migration script without using the `ATP` option and complete your migration, then later decide to migrate to ATP inventory after running your migrated system:

    Running the `migrateATP` script after migrating to WebSphere Commerce 5.4 generates the `ctos.sql`script. In this case, you need to run it before you restart the Web server and the `WebSphere Commerce Server — instance_name` application server.

  Note that this step is optional, as you may wish to implement your own tools and commands and thus not change the status to `S`.

  Although you will be able to view and edit the migrated orders and order items that are in the `C` state with the WebSphere Commerce 5.4 Commerce Accelerator tools, *it is not recommended that you edit them using these tools*. If you try to edit them using these tools (which you should not do as they are already completed), there may be nonrecoverable errors. In this case, the order status changes to either `E` state (CSR edit - a Customer Service Representative is working with the order) or `T` state (temporary - used by the Order Management user interface to temporarily back up an order). The CSR can view the summary of the `E` state order to find out the order number for the `T` state order. The `T` state order is the

backup copy of the original order (before editing with the tools). The CSR can print out this T state order for reference and using the Commerce Accelerator, manually recreate the order for the customer.

Note that the migration script does not associate the order items in the C state with a specified item (ITEMSPC) as this may degrade performance, as there may be large numbers (millions) of completed order items.

- Orders or order items are in the P state (pending - the customer can modify this order) or I state (submitted - the customer has submitted the order, but has not yet initiated payment).

  The migration script will unlock these orders by setting ORDERS.LOCKED to 0 (zero) for these orders.

Master catalog

In WebSphere Commerce Suite 5.1, the catalog system did not require structured catalog data. With the introduction of the master catalog, WebSphere Commerce 5.4 requires catalog data to be structured in a certain way.

In WebSphere Commerce 5.4, the master catalog is the central location to manage your store's merchandise. Everything you need in your store is contained within the master catalog. It is the single catalog containing all products, items, relationships and standard prices for everything that is for sale in your store.

Every store in the WebSphere Commerce system must have a master catalog. You can share the master catalog across stores and define as many stores as needed. In addition to creating a master catalog for catalog management, you may also choose to create one or more navigational catalogs for display purposes. A navigational catalog may contain the same entries as the master catalog, but with a much more flexible structure for customer display purposes. You can have as many navigational catalogs as you want. However, since it is the master catalog that is used to manage your online merchandise, it is recommended that you also use the master catalog as your navigational catalog to minimize maintenance overhead.

You can use the Product Management tool to view and manage your master catalog

For additional information on creating and managing a master catalog, see the WebSphere Commerce 5.4 online help after you have completed your migration.

Important structural considerations

In order to use the WebSphere Commerce 5.4 catalog tools such as the Product Management tool, your master catalog must meet the following conditions:

- The master catalog must be a proper tree, which means that there are no cycles. This implies that you must avoid the following type of scenario: The parent category **A** has a subcategory **B**. It is important that **B** and any of **B**'s subcategories are not the parent category of **A**.
- You cannot categorize a product under more than one category. To place a product in more than one category, use a navigational catalog.
- All items that belong to a product must be categorized under the same category as the product is categorized.
- The Product Management tools only work with a master catalog.

Migration of catalog information

If you are currently using one catalog in your Commerce Suite 5.1 store, the database migration script will assign that catalog as the master catalog for your store.

If you are currently using multiple catalogs in your Commerce Suite 5.1 store, you need to begin thinking about which one needs to be assigned as the master catalog for your migrated store. This assignment is done using the `choosemc.sql` script as described in "Assigning a master catalog" on page 89.

Default contract

> Business

WebSphere Commerce 5.4, has introduced terms and conditions to provide contract support. The migration process will create a default contract for your system which will have the same behavior and characteristics as the WebSphere Commerce Suite 5.1 business flows (for example, for shipping charges).

Normally, you do not need to take any actions during the migration process, since the default contract is automatically created. If you need to create additional contracts for your business processes, use the WebSphere Commerce Accelerator to do so after you have migrated to WebSphere Commerce 5.4. See the section "Creating a new contract" in the WebSphere Commerce 5.4 online help.

Access control

Access control is an integral part of the overall security and flow control for a commerce site. Site administrators and participating organizations need to control which participants can perform which actions on which objects in the system. In order to increase the customizability by externalizing the access control decisions from the code that manipulates business objects, WebSphere Commerce 5.4 resource level access control is policy-based instead of programmatic as in WebSphere Commerce Suite 5.1.

In WebSphere Commerce 5.4, access control is managed using a GUI and XML files which are used to define access control policies. These policies are stored in the WebSphere Commerce 5.4 database. They are loaded into memory when you startup the WebSphere Commerce 5.4 system.

Following are the key improvements made to access control in WebSphere Commerce 5.4:

- To increase flexibility, access control policies are externalized from code that manipulates business objects.
- Hierarchical access control is built into the access control model.
- All access control policies are based on groups: ActionGroups, ResourceGroups, AccessGroups.

If you are satisfied with the command level access control as implemented by Commerce Suite 5.1, and do not intend to make any code changes:

- You will need to migrate the Commerce Suite 5.1 access control tables to the appropriate policies. In order for this to work properly, you must run the database preparation script as described in "Running the database preparation script" on page 74. If this is not done, the access control policies for some customized commands and views will not be migrated.

- If you have overwritten the `getResourceOwners()` method, each of the resource owners returned will be used as the owner of the protectable resource, the command, for command level access control.
- If you have not overwritten the `getResourceOwners()` method, the owner of the protectable resource, the command, will be the store owner, if `storeId` is specified in the command context. Root organization will be used if no `storeId` is specified in the command context.
- If you have overwritten the method `checkPermission()`, this method will be called after command level access control is performed.

If you want to take advantage of the resource level access control in WebSphere Commerce 5.4, in addition to the above steps, you have to implement the `getResources()` method. In addition, if you are migrating any existing commands, you may want to remove your own implementation of `getResourceOwners()` if you have implemented it. Using the store owner or site organization should be adequate for command level access control. The finer level of access control can be performed by the resource level access control.

For more information on access control, see Chapter 8, "Access control subsystem considerations," on page 137.

Notes:

1. If you have added a controller command that extends from a Commerce Suite 5.1 command, WebSphere Commerce 5.4 will only add the command level policy for it during migration. If the WebSphere Commerce 5.4 version of the Commerce Suite 5.1 command implements `getResources()`, then you either have to determine what resources it is returning and create the appropriate resource level policy for your command, or, if you do not want resource level access control, you have to override `getResources()` on your command so that it returns a null value.

   To determine what the WebSphere Commerce 5.4 commands are returning for their `getResources()`, analyze the trace and look for `Action=WCBECommand` and find all the Protectable resources `getResources()` is checking for. In the trace above, the resource is Order.For example, consider if after enabling the SERVER trace, you find the following in the logs:

   ```
   ==============
   TimeStamp:    2001-11-16 02:42:30.937
   Thread ID:    <Worker#3>
   Component:    SERVER
   Class:        AccManager
   Method:       isAllowed
   Trace:        isAllowed? User=10012; Action=com.fvt.ACCOrderItemAddCmd;
   Protectable=com.ibm.commerce.order.objects._Order_Stub;
     Owner=7000000000000002000resource is Groupable

   ==============
   TimeStamp:    2001-11-16 02:42:30.984
   Thread ID:    <Worker#3> Component:
   SERVER Class: AccManager
   Method:       isAllowed
   Trace:        PASSED? =false
   ```

   The meaning of the above trace is that the resource level policy is failing. In this case, `ACCOrderItemAddCmd` is extending from the Server `OrderItemAdd` command which implements `getResources()`. Therefore, by default, `ACCOrderItemAdd` also requires a resource level policy, unless the

`getResources()` on it is changed to return null. This resource level policy is not added during migration, since it is not known what WebSphere Commerce 5.4 commands that you are extending.

In most cases, commands return access beans in the `getResources()` method. For example, returning `com.ibm.commerce.`*`xyz`*`.objects.`*`XYZ`*`AccessBean` in `getResources()` will appear as `com.ibm.commerce.`*`xyz`*`.objects._`*`XYZ_`*`Stub` in the trace. This difference is because WebSphere Commerce 5.4 must narrow the access bean to its remote interface (since it is the remote interface of the EJB that actually extends the Protectable interface).

2. In WebSphere Commerce Suite 5.1, resource level access control was enforced programmatically, within the command logic. In WebSphere Commerce 5.4, resource level access control policies are specified externally, similar to the way command level access control policies are specified. During the migration, only command level access control policies are migrated from Commerce Suite 5.1 to WebSphere Commerce 5.4. Any resource-level access control policies that are needed, due to customization of the Commerce Suite 5.1 default access control policies (which are stored in the `ACCCMDGRP` table), need to be added manually. Otherwise you will receive an unexpected access control violation exception. For details, see the related item in Appendix H, "Troubleshooting," on page 211.

Member subsystem

A key difference between WebSphere Commerce 5.4 and WebSphere Commerce Suite 5.1 is that in WebSphere Commerce 5.4, every user and organizational entity member must have a parent member which is another organizational entity. This allows users and organizational entities to form a membership hierarchy. Member groups are not part of the membership hierarchy and therefore they do not have parent members.

During the migration process, the database migration script determines the parent and ancestors for users and organizational entities based on the following:

- Whether users have records in the `BUSPROF` table and the values of the `ORG_ID` and `ORGUNIT_ID` columns.
- The value of the `MEMBER_ID` column in the `ORGENTITY` table for organizational entities.

As a result of determining the parent and ancestors of users and organizational entities, the `MBRREL` table is populated, which captures the membership hierarchy. WebSphere Commerce 5.4 business logic uses the membership hierarchy. Therefore, you should ensure certain columns in your database contain the proper values so that the membership hierarchy can be properly determined. The parent and ancestors of users and organizational entities are determined as follows by the database migration script:

- For users with a record in the `BUSPROF` table and with profile type set to B (B2B user):
  - The `ORGUNIT_ID` is used as the parent member ID if it is not null.
  - If the `ORGUNIT_ID` is null, the `ORG_ID` is used as the parent member ID if it is not null.
  - If both `ORGUNIT_ID` and `ORG_ID` are null, the `Default Organization` organizational entity (`ORGENTITY`) is used as the parent member.

To prevent your B2B users from being assigned the `Default Organization` as their parent in the migration process, you should scan the Commerce Suite 5.1 `BUSPROF` table and try to fill in the `ORGUNIT_ID` and `ORG_ID` columns prior to

migration. It is not recommended that `Default Organization` be the parent organizational entity of B2B users. If you cannot fill in the `ORGUNIT_ID` and `ORG_ID` columns in the `BUSPROF` table for certain registered users, you should change the profile type of those registered users from `B` (B2B user) to `C` (B2C user).

There is a `PROFILETYPE` column in the `USERS` table which allows null, `B`, or `C` as valid values:
– B (registered B2B user)
– C (registered B2C user)
– null (no profile data)

If you have customized Commerce Suite 5.1 code and this code has not been setting the profile type for your users, you should *clean up* the `PROFILETYPE` column in the `USERS` table. For WebSphere Commerce 5.4, it is recommended that:
– B2C users be under `Default Organization` and have profile type `C`. Typically, B2C users do not have records in the `BUSPROF` table and have the `Default Organization` as their parent.
– B2B users have profile type `B`, have records in the `BUSPROF` table, and have the appropriate organizational entity as their parent. It is not recommended that B2B users be under `Default Organization`.

In addition, ensure that you set your administrators (that is, those users whose register type in the `USERS` table is an `A` or `S` and that have entries in the `ACCMBRGRP` table) to have a profile type of `B`.

- Users without a record in the `BUSPROF` table, will have their parent organizational entity set to the `Default Organization` by the database migration script.
- For organizational entities in the `ORGENTITY` table, the `MEMBER_ID` column is used as the parent member ID, if it is not null. If the `MEMBER_ID` column is null, the parent member is set to be the `Root Organization`.

  With the new `MBRREL` table which contains records for registered users and organizational entities, the database migration script determines membership hierarchy solely using the content of the `MBRREL` table. Note that the `MBRREL` table does not contain records for the generic user, guest users, and member groups.

  During migration, the following consistency check is performed by the database migration script: For a user with a record in `BUSPROF`, if the `ORG_ID` and `ORGUNIT_ID` columns are not null, then starting with `ORGUNIT_ID`, the script uses the `ORGENTITY` table to traverse up the organizational hierarchy until `MEMBER_ID` eventually contains a null or contains the same value as `ORGENTITY_ID`. It checks that the value of `MEMBER_ID` column in the `ORGENTITY` table is the same as the `ORG_ID` column in the `BUSPROF` table. If it is not, the script notifies the user to correct the inconsistency.

Other member subsystem considerations
- During migration to WebSphere Commerce 5.4, the `STATE` column in the `MEMBER` table gets set as follows:
  – For guest users (where register type is `G`), the migration script sets `STATE` to null.
  – For registered users (where register type is `R`), the migration script sets `STATE` to `approved`.
  – For organizational entities, the migration script sets `STATE` to `approved`.
  – For member groups, he migration script sets `STATE` to null.

Organizational entities are migrated so that they do not own any approval groups, which means by default, the migrated organizational entities do not want approvals for business processes such as B2B users self-registration.

- The roles that are shipped by default in WebSphere Commerce 5.4 include all of the roles shipped in Commerce Suite 5.1 plus some new roles, with the following exceptions:

  – The Order clerk role is not included.

    Commerce Suite 5.1 supported an Order Clerk role which is no longer required or used in WebSphere Commerce 5.4. The tasks that used to be performed by the Order Clerk role have either been automated or can be performed by the Customer Service Supervisor in WebSphere Commerce 5.4. If a user has an Order Clerk role (-5) in Commerce Suite 5.1 and an entry in the ACCCMDGRP table, the user is migrated as part of the access control migration and the role is treated as a user-defined role.

    Review if you still require the Order Role clerk. If you do not require it, remove it since it is no longer supported in WebSphere Commerce 5.4.

  – The Customer role is not included.

    Commerce Suite 5.1 included an access group called Customer. Every access group in Commerce Suite 5.1 has a name which is a role name. Normally, an access group in Commerce Suite 5.1 is used for two purposes:

    - Assigning commands to the access group (association stored in the ACCCMDGRP table)
    - Assigning users to the access group (association stored in the ACCMBRGRP table)

    Users assigned to an access group play the role with a name same as the access group name. Thus, assigning a user to an access group is the same as assigning a role to that user, allowing the user to execute commands associated with that access group. The Customer access group in Commerce Suite 5.1 is associated with a set of commands that *all users* can execute. This means the Customer access group represents all users in the Commerce Suite 5.1 system, and it is not necessary to assign every user with the Customer role.

    In WebSphere Commerce 5.4, an AllUsers member group is shipped to replace the Customer access Group. An access control policy has been created to associate the set of commands that can be executed by all users with the AllUsers member group. Since explicit assignment of every user with a Customer role is unnecessary, a Customer role is not shipped by default in WebSphere Commerce 5.4. For a particular organization entity, there is a set of users who are employees of the organizational entity, and the non-employees are considered *customers*. Users that have been explicitly assigned into the Customer access group in Commerce Suite 5.1, are explicitly assigned into the AllUsers member group during data migration to WebSphere Commerce 5.4. In such cases, the migration script issues a warning message, since such explicit assignment is not necessary.

  – The Merchant role has been renamed to Seller (*Merchant* is a B2C term, and *Seller* is a B2B term).

  – The Merchandising Manager role has been renamed to Product Manager for the same reason.

- In Commerce Suite 5.1, the USERS table contains a REGISTERTYPE column which accepts four valid values:

  – R — registered users

– `G` — guest users
– `S` — Site Administrator
– `A` — Administrator

Register types `S` and `A` are role-related. In WebSphere Commerce 5.4, the `REGISTERTYPE` column in the `USERS` table continues to support the same set of values. However, given the set of roles that are available by default from WebSphere Commerce 5.4, the meaning of register type `A` needs to be refined. Type `A` is now given a more concrete meaning of: *employees of the Seller organization who play certain roles*. The set of roles which correspond to register type `A` is configurable by changing the definition of the Administrators access group using the Administration Console. In addition, the register type value of a user is set to an `A` or an `S` automatically during role assignment and unassignment, thus ensuring the value of the register type is consistent with the roles played by a user.

In summary:
– If a user from the Seller organization is assigned the Site Administrator role, the register type value for this user is `S`
– If a user from the Seller organization is assigned one of the roles defined in the Administrators access group other than Site Administrator, the register type value for this user is `A`.

- During migration, the database migration script performs the following. The script explicitly assigns users from Commerce Suite 5.1 with register type of `A` and that do not belong to any access group, to the Administrators access group in WebSphere Commerce 5.4. Note that if there are no entries in the `ACCMBRGRP` table, then the script does not perform this step. Before running the migration script, you should do the following:
   – Examine the definition of the Administrators access group and modify the list of roles if necessary. For example, if you have assigned users in Commerce Suite 5.1 to an access group named XXX with `REGISTERTYPE=A` and XXX is not one of the roles listed in the Administrators access group in WebSphere Commerce 5.4, you should add `role=XXX` as an additional criteria to the Administrators access group.
   – If you have any customized logic which directly examines the `REGISTERTYPE` column of the `USERS` table for the `S` and `A` values, you should change the code as follows:
      - Replace your logic with an access control policy if your logic tries to examine the `REGISTERTYPE` column in order to determine authorization to be given to a user. In WebSphere Commerce 5.4, a new access control design is available and it is recommended that you use access control policies instead of hard coding authorization-related logic. See "Access control" on page 9.
      - If your logic examines the `REGISTERTYPE` column, but not for access control purposes, you should use one of the programming interfaces available from the Member Subsystem:
         - `isAdministrator()`
         - `isSiteAdministrator()`
         - `isMemberInRole()`

      See the WebSphere Commerce 5.4 online help after you have migrated for more information on these interfaces. This change removes any dependency

your code has on the actual values of register type. Note that in a future version of WebSphere Commerce, the set of values that are valid for register type may change.

– During migration from Commerce Suite 5.1 to WebSphere Commerce 5.4, the EXCLUDE column in the MBRGRP table for migrated users within a member group is set to 0 (zero). A value of zero means such users are explicitly included in the member group.

Minimizing downtime during migration

This section describes a high-level approach to minimize the time your system is offline during your migration using two machines.

If you have extensively customized your Commerce system, you may need to consult IBM support to use this approach.

The migration process described in this Migration Guide is generally for a migration *in-place*. That is, you migrate your Commerce Suite 5.1 system to a WebSphere Commerce 5.4 on the same machine. This requires that you shut down your Commerce Suite system during the migration process until you can bring your WebSphere Commerce 5.4 online.

The following sections provide two approaches to minimize the time that your system is off line due to the migration progress. Both these approaches require two machines.

Note: These approaches, while minimizing the time your Commerce Suite system is down, will require additional hardware resources to stage the WebSphere Commerce 5.4 system. However, after migration is complete, you will be able to redeploy the Commerce Suite machine for other uses.

Installing WebSphere Commerce 5.4 on a separate machine

In this approach, you install WebSphere Commerce 5.4 on a new machine and copy your WebSphere Commerce Suite 5.1 assets on the new machine, and migrate those assets

1. Back up your Commerce Suite 5.1 system, and your Commerce Suite 5.1 and WAS databases as documented in Chapter 2, "Backing up Commerce Suite 5.1," on page 27.

2. Install WebSphere Commerce 5.4 on a new machine that meets the requirements for this product as documented in the *WebSphere Commerce 5.4 Installation Guide* for your platform and edition.

Install to the same installation directory as you have used in Commerce Suite 5.1 to avoid having to manually update path names in the configuration files later. (The default installation directory for Commerce Suite 5.1 is /usr/lpp/CommerceSuite.) Do *not* create a WebSphere Commerce 5.4 instance.

This book generally refers to the WebSphere Commerce 5.4 default installation directory, /usr/WebSphere/CommerceServer. If you do install WebSphere Commerce 5.4 to the same directory as WebSphere Commerce Suite 5.1, note that your installation directory will be different from those referenced throughout the book.

> **Important**
>
> + After you install WebSphere Commerce 5.4, you need to install a
> + migration bundle that is available from the WebSphere Commerce
> + Support site (`http://www.ibm.com/software/commerce/support/`). Click
> + on the link for the edition you are using and see the section **Software**
> + **downloads**. Download the migration bundle, APAR IY46716 and follow
> + the instructions that are provided in the bundle to install it.
> +
> + You should not proceed with the migration steps in this Guide (such as
> + migrating your instance or database) without installing this migration
> + bundle.

3. Restore the Commerce Suite database (which you backed up in step 1 on page
   15) to the WebSphere Commerce 5.4 system. See the product documentation
   supplied with your database (either DB2 or Oracle) on how you restore a
   database. You should not need to restore the `WAS` database as that database
   applies to IBM WebSphere Application Server 3.5 and you create a new `WAS40`
   database when you install WebSphere Application Server 4.0.2.

4. Copy the Commerce Suite instances directories to the WebSphere Commerce
   5.4 machine. For example, using the default instance location for Commerce
   Suite 5.1, you need to copy all your
   `/usr/lpp/CommerceSuite/instances/`*`your_instance`* directories from your
   Commerce Suite 5.1 machine to your WebSphere Commerce 5.4 machine.

   Note: If you have installed WebSphere Commerce 5.4 to a different location
   than Commerce Suite 5.1, you will have to locate and update any
   references to path names in the configuration files.

5. Modify your *`instance_name`*`.xml` (for example, `demo.xml`) and other related
   `.xml` files and change all occurrences of the old host name to new host name
   of the WebSphere Commerce 5.4 machine. The following is a list of the files
   that need to be updated:
   - *`instance_name`*`.xml`
   - *`instance_name`*`_was.xml`
   - *`instance_name`*`_was_DB.xml`
   - *`instance_name`*`_was_EJB.xml`
   - *`instance_name`*`_was_Start.xml`
   - *`instance_name`*`_was_Stores.xml`
   - *`instance_name`*`_was_Tools.xml`
   - *`instance_name`*`_was_VH.xml`

6. Copy the following files from the Commerce Suite 5.1 machine to the
   WebSphere Commerce 5.4 machine:
   - The `wcs_instances` file in the `/usr/lpp/CommerceSuite/instances` directory
     to the corresponding `instances` directory where you have installed
     WebSphere Commerce 5.4.
   - The `cfg.passwd` file in the `/usr/lpp/CommerceSuite/bin` directory to the
     corresponding `bin` directory where you have installed WebSphere
     Commerce 5.4.

7. Update your WebSphere Commerce `product.xml` file as described in
   "Updating product.xml" on page 66. If you are migrating from WebSphere

Commerce 5.1.1.2, you also need to update the `migrate.xml` file as described in "Updating migrate.xml (when migrating from WebSphere Commerce 5.1.1.2)" on page 67.

8. Run the data migration procedure as documented in Chapter 4, "Migrating your database," on page 69 on your restored Commerce Suite database on the WebSphere Commerce 5.4 machine.

9. Run the instance migration procedure as documented in "Migrating your Commerce Suite 5.1 instance configuration" on page 99.

10. Run the merchant key migration procedure as documented in "Migrating your security configuration" on page 104.

11. Migrate your store data as documented in "Migrating store file assets" on page 109.

12. Perform all the other migration procedures documented in this Guide, as required.

13. Verify that you can publish and shop at your WebSphere Commerce 5.4 store, and that your system is generally operational.

14. Enhance your WebSphere Commerce 5.4 system as required in order to take advantage of the new features. See the WebSphere Commerce 5.4 online help for details on how to implement any new feature.

15. Refresh your WebSphere Commerce data with up-to-date information from your production Commerce Suite 5.1 system, that is still online:

    a. Backup your migrated WebSphere Commerce 5.4 database.

    b. Export the migrated USERREG, PATTRVALUE, ORDPAYINFO, and ORDPAYMTHD tables in your WebSphere Commerce 5.4 database to a text file. The tables will contain the migrated passwords and credit card information for users of your WebSphere Commerce system.

    c. Take your Commerce Suite 5.1 system offline.

    d. Backup the Commerce Suite 5.1 database and restore it to your WebSphere Commerce 5.4 machine once again, in order to capture changes to the Commerce Suite 5.1 database since the first backup and restore.

    e. Shut down your Commerce Suite 5.1 system.

    f. Run the data migration procedure as documented in Chapter 4, "Migrating your database," on page 69 on the up-to-date Commerce Suite 5.1 database.

    g. If new users or new credit card information *have not been* added to your system while you are refreshing your database, import the USERREG, PATTRVALUE, ORDPAYINFO, and ORDPAYMTHD tables from the text file in step 15b. These tables will contain the migrated passwords and credit card information for users of your WebSphere Commerce system.

       If new users or new credit card information *have been added* to your system while you are refreshing your database using the previous steps:

       • Copy the original merchant key in the Commerce Suite 5.1 *instance*.xml file into the migrated WebSphere Commerce 5.4 *instance*.xml file.

       • Remigrate your password and credit card information as described in "Migrating your security configuration" on page 104.

16. After verifying that your WebSphere Commerce 5.4 system is operational, you can bring it online.

Duplicating your WebSphere Commerce Suite 5.1 environment on a separate machine

In this approach, you duplicate your WebSphere Commerce Suite 5.1 environment on a new machine, and migrate those assets to the WebSphere Commerce 5.4 level *in-place* on the new machine:

1. Back up your production Commerce Suite 5.1 system, and your Commerce Suite 5.1 and WAS databases as documented in Chapter 2, "Backing up Commerce Suite 5.1," on page 27. Ensure the following items are accessible:

    -   The WebSphere Commerce Suite 5.1 database image
    -   The WebSphere Commerce Suite 5.1 web asset files (for example, JSPs and *.html. files).
    -   Any customized properties files.

2. Install WebSphere Commerce Suite 5.1 to a new machine which will eventually be your WebSphere Commerce 5.4 machine, which for the purpose of this discussion will be called the *staging machine*.

3. Restore the database image and the web assets files that you backed up in 1 to the staging machine.

4. Set up the staging machine environment and configuration to be as close as possible to your production machine. Ensure that WebSphere Commerce Suite 5.1 system and your stores are operational on the staging machine.

5. Back up the staging server machine as described in in Chapter 2, "Backing up Commerce Suite 5.1," on page 27.

6. Proceed with the *in-place* migration on the staging machine as described in this Guide. In particular:

    -   Upgrade your software levels to that required for WebSphere Commerce 5.4, as documented in Chapter 3, "Upgrading your software," on page 35.
    -   Run the data migration procedure as documented in Chapter 4, "Migrating your database," on page 69.
    -   Run the instance migration procedure as documented in "Migrating your Commerce Suite 5.1 instance configuration" on page 99.
    -   Run the merchant key migration procedure as documented in "Migrating your security configuration" on page 104.
    -   Migrate your store data as documented in "Migrating store file assets" on page 109.
    -   Perform all the other migration procedures documented in this Guide, as required.

7. Verify that you can publish and shop at your migrated store, and that your system is generally operational.

8. Once it is operational at the WebSphere Commerce 5.4 level, enhance your staging system as required in order to take advantage of the new WebSphere Commerce 5.4 features. See the WebSphere Commerce 5.4 online help for details on how to implement the new features.

9. Refresh the WebSphere Commerce data on your staging machine with up-to-date information from your production Commerce Suite 5.1 system, that is still online:

    a. Backup your migrated WebSphere Commerce 5.4 database on your staging machine.

    b. Export the migrated USERREG, PATTRVALUE, ORDPAYINFO, and ORDPAYMTHD tables in your WebSphere Commerce 5.4 database to a text file. The tables

will contain the migrated passwords and credit card information for users of your WebSphere Commerce system.

  c. Take your Commerce Suite 5.1 system offline.

  d. Backup the Commerce Suite 5.1 database and restore it to your WebSphere Commerce 5.4 staging machine once again, in order to capture changes to the Commerce Suite 5.1 database since the first backup and restore.

  e. Shut down your Commerce Suite 5.1 system.

  f. Run the data migration procedure as documented in Chapter 4, "Migrating your database," on page 69 on the up-to-date Commerce Suite 5.1 database.

  g. If new users or new credit card information *have not been* added to your system while you are refreshing your database, import the USERREG, PATTRVALUE, ORDPAYINFO, and ORDPAYMTHD tables from the text file in step 9b on page 18. These tables will contain the migrated passwords and credit card information for users of your WebSphere Commerce system.

If new users or new credit card information *have been added* to your system while you are refreshing your database using the previous steps:

- Copy the original merchant key in the Commerce Suite 5.1 *instance*.xml file into the migrated WebSphere Commerce 5.4 *instance*.xml file.

- Remigrate your password and credit card information as described in "Migrating your security configuration" on page 104.

10. After verifying that your WebSphere Commerce 5.4 system is operational, you can bring it online.

Transitioning to WebSphere Application Server 4.0

A major transition in WebSphere Commerce 5.4 from WebSphere Commerce Suite 5.1 is its support for WebSphere Application Server 4.0. This section provides an overview of some of the major items you need to consider prior to making the transition to WebSphere Application Server 4.0.

Version 4.0 of WebSphere Application Server is fully compliant with Java 2 Platform, Enterprise Edition (J2EE) specifications, which has caused many changes in the organization of the WebSphere Application Server product relative to that of Version IBM WebSphere Application Server 3.5.

The detailed steps to upgrade to this level of WebSphere Application Server is described in "Upgrading to WebSphere Application Server 4.0.1" on page 39.

Transition overview

The following sections highlight the major changes in WebSphere Application Server 4.0 from earlier versions.

- J2EE creates a clear separation between development (creating the application) and administration (installing and managing the application).

This separation enables the development of applications that are independent from the environments in which they are deployed. In addition, J2EE task separation simplifies the process of promoting an application from initial development up through production, or of moving an application from one server to another. In each of these cases, changes to application code are not necessary; only deployment parameters might change.

Version 4.0 supports J2EE task separation through reorganized interfaces. In Version 3.x, developers used the administrative console to create, edit, and view

applications. In Version 4.0, developers use the Application Assembly Tool (AAT) to create, edit, and view J2EE applications.

In Version 4.0, each application is installed into the server domain and bound to an environment when the application is installed. This enables administration at the application and module level. Administrators no longer need to manage individual servlets, JSPs, or beans.

- The relationship between applications and application servers has changed in J2EE.

An enterprise application can contain many Web modules and EJB modules. Each module can be installed onto a different application server or server group, even if the servers or server groups are on multiple nodes. As a result, a single application can contain many modules, spread over many application servers or server groups. Likewise, a single application server or server group can have modules from many different applications installed on it.

After a J2EE application is created, you install it onto application servers through the administrative console. Through the administrative console, you can view installed modules either by the application to which they belong or by the application server on which they are installed. Modules can be started and stopped individually or collectively. Modules can be started collectively by either starting the application to which they belong or by starting the application server on which they are installed. Modules can be stopped in a similar way.

Deploying new J2EE applications

There are two steps involved in creating J2EE applications: copying the appropriate files into the archive (classes, JSPs, HTML, image files, and so on) and creating deployment descriptor files for the modules and applications. In Version 4.0, the AAT supports both steps by enabling users to copy files with appropriate relative paths into the archive, as well as by providing a GUI method for defining deployment descriptors.

Developers can also set environment-specific binding information through the AAT. These bindings are used as defaults when the application is installed through the administrative console. In addition, users can define IBM extensions to the J2EE specification, such as allowing servlets to be served by class name. To ensure portability to other application servers, these extensions are saved in a separate XML file from the standard J2EE deployment descriptor.

Role-based security

Version 4.0 security is consistent with J2EE role-based security specifications. Roles are specified in the deployment descriptors for an application; these roles are then bound to users or to groups when the application is installed. In the administrative console, a Security Center enables you to perform all security tasks from a single location. This encompasses everything from changing the binding information for roles in an application to setting SSL properties to enabling security. Application-specific security tasks can also be done through the property sheets for each application.

Redeployment of previously installed applications

In Version 3.x, all tasks were performed through the administrative console. In Version 4.0, application settings are defined in J2EE deployment descriptors through the AAT.

Unless you must change information that affects the bindings of an installed application, you can edit and save the deployment descriptors in place. To redeploy such an application, open the AAT directly on the `installedApps` folder that holds the application.

You can also create or edit applications manually. For example, if you need to add a JSP or change a servlet class, you can simply place the new or changed file in the appropriate location in the `installedApps` folder.

To redeploy an installed application that requires changes to binding, you export the application through the administrative console, edit the application in the AAT, and reinstall the application through the administrative console. Because existing binding information is saved during the export step, the only additional binding information needed is for the components or modules added during editing.

Note: For security and consistency, Web application URLs are now case-sensitive on all operating systems.

Support for J2EE resource types

In addition to JDBC providers and datasources, several resource types were added in Version 4.0: URLs, JMS, and JavaMail. In each case, you can create a resource provider (JDBC providers, URL providers, and JMS providers) and then create resource factories for each provider (datasources, URLs, JavaMail sessions, JMS destinations, and JMS connections). In the case of JavaMail, the default JavaMail provider is not shown in the administrative console, because it is not configurable and additional JavaMail providers cannot be created.

J2EE impact to models and cloning

In Version 3.x, many different types of objects can be modeled and cloned. With the switch to J2EE compliance in Version 4.0, only application servers can be cloned. These models are now called server groups, and each server group can contain multiple application servers, or clones.

Where to go next for more information

For more information about J2EE, visit the following Web site:

`http://java.sun.com`

For more information about changes in configuration support, see the related information in the WebSphere Application Server 4.0 migration information. WebSphere Application Server 4.0 information is available from the WebSphere Application Server Info Center at the following Web site:

`http://www.ibm.com/software/webservers/appserv/infocenter.html`

For information about how to upgrade to Version 4.0, see the sections below.

Migrating product prerequisites

The WebSphere Application Server 4.0 prerequisites Web page contains up-to-date information about the supported prerequisites and corequisites:

`http://www.ibm.com/software/webservers/appserv/doc/v40/prereqs/ae_v401.htm`

Be sure to check whether your JDBC provider is at the right level for the new installation. This driver will be required by the product administrative server in order to connect to its administrative database.

Migrating DB2, IBM HTTP Server, and other complimentary prerequisites

WebSphere Application Server 4.0 simplifies the migration of product prerequisites by providing the option to install a complimentary Web server, database, and IBM SDK for Java on your supported operating system. IBM SDK for Java is the exact

level and type needed by WebSphere Application Server 4.0. See the WebSphere Application Server 4.0 installation guides for further details.

The compact disc version of the product includes the complimentary prerequisites; Web download versions can vary (offered with and without database, and so on), to provide a choice of download file sizes. If you are not installing from the CD, consult the product Web site for details. Make sure you download the installation package with the features you want.

You can uninstall the back-level prerequisites and install brand-new versions when you install the product.

Migrating non-IBM prerequisites

Some prerequisite or corequisite products, such as an Oracle database, are not provided as part of the WebSphere Application Server 4.0 installation. To upgrade Oracle, the best source of information is the Oracle product documentation.

First, consult the previously cited prerequisites page to determine which software requires migration or upgrade. Second, consult the documentation for the particular products to learn how to migrate to the version supported by this product. For prerequisites not offered during the WebSphere Application Server 4.0 installation, the safest approach is to migrate or upgrade prerequisites *before* installing WebSphere Application Server 4.0.

Mapping configurations to Version 4.0

This section details how objects and attributes are mapped to the Version 4.0 environment when you restore a configuration from an earlier product version.

- Directories `stdin`, `stdout`, and `stderr`; passivation directory and working directory.

  Because the typical location for these directories might include Version 3.x installation directories and the location might be different in the new Version 4.0 installation, additional checking is done for these entries if they are specified. Changed from Version 3.x, the default location for `stdin`, `stdout`, and `stderr` is the `logs` directory in Version 4.0 installations. Existence of the passivation and working directories is checked before the directories are mapped. If they exist, they are used; otherwise, appropriate defaults are used instead.

- Enterprise beans

  EJB 1.0 was the only specification level supported in Version 3.x; EJB 1.1 is the only level supported in Version 4.0. However, many EJB 1.0 beans can be deployed as EJB 1.1 beans successfully. Enterprise beans are redeployed automatically as part of the application migration phase. Be sure to check WASPostUpgrade.log for details of the deployment of these beans; make required changes and redeploy.

- JDBC providers and datasources

  JDBC and DataSource objects are significantly redefined in Version 4.0. These objects are mapped to the new configuration by using the Version 3.x settings as input variables.

  You might notice a difference between the datasources mapped from Version 3.x and those defined by the samples. The difference is in the user ID and password fields of the datasource. The samples provide a default user ID and password, but the migrated datasources do not. This is because user ID and password data

is defined in enterprise-bean bindings, not in the datasource. In Version 3.x, the information is defined at the container and EJB level and so must be mapped to the enterprise bean.

- JSP levels

  JSP 0.91 is not supported in Version 4.0. JSP objects that are configured to run as JSP 0.91 are not migrated, but they are noted in the output and logged. JSP 1.0 and 1.1 objects are run as JSP 1.1, because that is the only supported JSP level in Version 4.0.

- Models and clones

  Models and clones have been dramatically redefined in Version 4.0. Application servers are the only objects supported as models and clones in Version 4.0. This is a significant difference from Version 3.x, in which many objects could be models and clones. All models and clones relating to application servers are mapped to server groups in Version 4.0.

  During the migration of all other objects that could be cloned previously, special mapping occurs. All clones are treated as simple objects and are mapped as if they were not clones. Models that are not application server models are ignored and not mapped.

- Multiple application servers

  In Version 4.0 Advanced Single Server and Advanced Developer editions, only one application server is configured at one time. In Version 3.x, there can be many application servers defined at one time. During migration of these objects to one of these Version 4.0 editions, the names of the application servers determine how migration occurs. If the names of the application servers match (for example, Default Server), the attributes of the Version 4.0 object are updated to match the previous configuration, and all children are migrated into that application server. If the names do not match, just the children of that Version 3.x application server are migrated to the one application server in the Version 4.0 environment.

- Node name

  A Version 3.x repository can contain more than one node name and its associated children. The `WASPostUpgrade` tool processes only those objects and children that match the node that is being migrated. This determination is made by checking the names of nodes in the configuration files with fully qualified and non-qualified network names of the machine that is being migrated.

- Servlet Redirector

  Servlet Redirector is not supported in Version 4.0; these objects are ignored.

- Transports

  The default transport type of the Servlet Engine in Version 3.x was Open Servlet Engine (OSE). Because OSE transport is no longer supported in Version 4.0, these transports are mapped to HTTP transports using the same port assignments.

- datasources.xml

  In Version 3.x, a `datasources.xml` file could be used to augment datasource configuration settings. This file was stored in the properties directory. If this file exists, the properties in the file are merged into the configuration of the datasource and JDBC provider configuration.

Restoring the previous configuration to the new installation

A set of migration tools is provided with the product to help migrate system configurations for installations at Version 3.x and later. The product installation program calls these tools as part of automated migration support. You can call them yourself from the command line.

The tool that restores the Version 3.x configuration is called `WASPostUpgrade`. This tool uses the information created by the `WASPreUpgrade` tool to restore the previous Version 3.x configuration to a Version 4.0 installation.

Because the Version 4.0 product adheres to the J2EE programming model and earlier versions do not, significant changes are required to apply the Version 3.x configuration to a Version 4.0 installation.

Creating and deploying J2EE applications

The J2EE programming model specifies an architecture for how applications are created and deployed. Because applications in Version 3.x were not architected in the same manner, the migration process recreates these applications. All migrated Web resources and enterprise beans are created in J2EE applications. All enterprise applications defined in the Version 3.x installation are mapped into J2EE applications with the same name and deployed in the default server. All other Web resources and enterprise beans that are mapped but not included in an enterprise application are mapped into a default J2EE application called *DefaultApplication*.

Web applications are mapped to J2EE WAR files. Enterprise beans are deployed as EJB 1.1 beans in J2EE JAR files. These resources are combined in a J2EE EAR file and deployed in the Version 4.0 configuration. There are some differences between the EJB 1.0 and EJB 1.1 specifications, but in most cases, EJB 1.0 beans can run successfully as EJB 1.1 beans. It is recommended that you carefully analyze `WASPostUpgrade.log` (see the end of this topic), because detailed information specific to each bean deployed is saved in the log.

Security

Security settings that were applicable in the Version 3.x environment are applied to J2EE security attributes as part of the migration process.

Samples

Samples are not migrated; they have been updated specifically for J2EE in Version 4.0. The new samples should be used instead of the ones previously provided with Version 3.x product.

Mapping details

See the Related information for more specific information about how objects and attributes are mapped to the Version 4.0 configuration.

WASPostUpgrade parameters

If the configuration is being restored on an Advanced Edition installation, the administrative server must be running for this command to run successfully.

```
com.ibm.websphere.migration.postupgrade.WASPostUpgrade backupDirectoryName
 [-import xml_data_file]
 [-adminNodeName primary_node_name]
 [-nameServiceHost host_name [ -nameServicePort port_number]]
 [-traceString trace_spec [-traceFile file_name]]
 [-substitute "key1=value1[;key2=value2;[...]]"]
```

The first parameter of the command is required. The others are optional, except as noted. A summary follows:

- *backupDirectoryName* - The name of the directory that contains the saved configuration and files created by the `WASPreUpgrade` command. This is a required parameter.
- [-import *xml_data_file*] - This optional parameter can be used to specify an XML data file that was created by using the Version 3.x XMLConfig tool. If this parameter is not specified, the default XML configuration file (`websphere_3x_backup.xml`) in the backup directory is used.
- [-adminNodeName *primary_node_name*] - The name of the administrative node for the current installation. This parameter is required if the configuration is being restored on an Advanced Edition installation. `XMLConfig` is called with this parameter.
- [-nameServiceHost *host_name* [-nameServicePort *port_number*]] - If specified, these optional parameters are passed to the `XMLConfig` call. They can be used to override the default host name and port number used by `XMLConfig`. These parameters are used only if the configuration is being restored on an Advanced Edition installation.
- [-traceString *trace_spec* [-traceFile *file_name*]] - These optional parameters are used to gather trace information for use by IBM service personnel. The value of traceString is "`*=all=enabled`" and must be specified with quotation marks to be processed correctly.
- [-substitute *"key1=value1[;key2=value2;[...]]"*] - If specified, this optional parameter is passed to the `XMLConfig` call. It is used for substitution of security values in the XML data file. In the input XML file, each key should appear as `$key$` for substitution.

Logging

The `WASPostUpgrade` tool displays status to the screen while it is running. `WASPostUpgrade` also saves a more extensive set of logging information in the logs directory. The file, `WASPostUpgrade.log`, can be viewed with a text editor.

Chapter 2. Backing up Commerce Suite 5.1

Before you proceed with the migration process documented in this Guide, you should perform a complete system backup of your working Commerce Suite 5.1 system. This will enable you to recover your previous system in the event that you encounter problems while migrating to WebSphere Commerce 5.4.

If you are migrating *in-place*, you cannot revert to your previous version of Commerce Suite 5.1 after you complete the migration to WebSphere Commerce 5.4. Coexistence of WebSphere Commerce 5.4 along with Commerce Suite 5.1 on the same machine is not supported.

Backing up your Commerce Suite 5.1 system

To back up your Commerce Suite 5.1 system:
- Perform a complete system backup of your Commerce Suite system in accordance with the documentation provided with your operating system or any specialized backup and restore software you are using. Typically, you can backup your system to a tape drive, a ZIP drive, or some other file system.
- The backup should include any customized files and directories used for Commerce Suite 5.1 and its associated components such as the database, the Web server, WebSphere Payment Manager, WebSphere Application Server and the IBM Developer Kit, Java 2 Technology Edition.
- In particular, backup the main Commerce Suite 5.1 installation directory including all underlying subdirectories and files to a location that is easily accessible during the migration process as you may need to refer to these directories and files at some point in the migration process.

Backing up directories and files

To manually back up any of the following directories or files:
1. From a command prompt, create a temporary backup directory.
2. Switch to the Commerce Suite 5.1 installation directory (/usr/lpp/CommerceSuite) .
3. Select the appropriate directories or files and copy them to the temporary backup directory.

In particular, you should backup the following directories and files:
- The Commerce Suite 5.1 installation directory:

  `/usr/lpp/CommerceSuite`
- The following subdirectories of the main Commerce Suite 5.1 installation directory:
  - `stores`
  - `web`
  - `instances\`*`your_instance`*

    (Or wherever the instances that were created are located.)
- The following important Commerce Suite 5.1 configuration files:
  - The `cfg.passwd` file in the `/usr/lpp/CommerceSuite/bin` directory.
  - The `wcs_instances` file in the `/usr/lpp/CommerceSuite/instances` directory.

- The httpd.conf file in the /usr/HTTPServer/conf directory or equivalent configuration files for other Web servers.
- The keyfile.kdb and keyfile.sth test key files in the /usr/HTTPServer/ssl path.
- The file wcs.server located in the /usr/lpp/CommerceSuite/instances/*your_instance*/xml/rules directory.
- Any file-based content such as static HTML pages and GIF files.
- Any database .sql scripts.
- Any JavaServer Pages (JSP files).
- Any customized commands and files (for example, .java, .class, .jar, .zip, or .properties files).
- Any customized document files (for example, .pdf or text files).
- The rules projects for campaigns. These files are generated when you publish a campaign. They are named after the campaign, but have a number of different extensions as follows:
  - *campaign_name*.adv
  - *campaign_name*.cdd
  - *campaign_name*.dbcp
  - *campaign_name*.flow0
  - *campaign_name*.flow1
  - *campaign_name*.jcp
  - *campaign_name*.rb

Once you start to run the campaign code in WebSphere Commerce 5.4, WebSphere Commerce will look at these files and store them persistently into the new WebSphere Commerce schema tables. Once the data gets transferred from these files to the schema, the campaign code will start to look at the database instead of the files. The files will eventually become obsolete when the campaign is run to completion in WebSphere Commerce 5.4.

Backing up your databases

The following sections describe how to backup your DB2 or Oracle® databases.

DB2 databases

To backup your DB2 database, perform the following actions from your DB2 server machine in a 2–tier environment (where your database is installed on a machine that is remote from Commerce Suite), or from your Commerce Suite machine in a single–tier environment (where your database is installed on the same machine as Commerce Suite):

1. Logon to your Windows® system as the Windows user ID that created the DB2 database, or that owns the DB2 database.
2. Log on as your DB2 instance owner ID (for example, db2inst1):

   su - *db2_instance*
3. Complete all database transactions.
4. Ensure that all applications are disconnected from each database. To view a list of all the applications that are connected to a database, run the following command:

   db2 list applications

   If all applications are disconnected, the following message displays:

```
SQL1611W No data was returned by the Database System Monitor. SQLSTATE=00000
```

To force all applications to disconnect, run the following command:

```
db2 force applications all
```

5. Ensure that all applications are stopped by reissuing the following command:

```
db2 list applications
```

If all applications are stopped, the following message displays:

```
SQL1611W No data was returned by the Database System Monitor. SQLSTATE=00000
```

6. Ensure that all your Commerce Suite 5.1 databases (such as MSER, MALL, and so on) are cataloged. To view a list of all the cataloged databases in the current instance, type the following command:

```
db2 list database directory
```

7. Create a directory where you will back up your databases, for example, /db2_backup.

8. Back up all Commerce Suite 5.1 databases (such as MSER, MALL, and so on) by typing the following command:

```
db2 backup database db_name to backup_directory
```

where *db_name* is the name of the database, and *backup_directory* is the full path to where you want to back up the database. The *backup_directory* must exist. You should receive a message indicating that the back up was successful. For example, to back up the MALL database to the above backup directory, use the following command:

```
db2 backup database MALL to /db2_backup
```

If you have more than one database, repeat the command for each Commerce Suite 5.1 database.

> You should consider backing up any non-Commerce Suite databases, such as the WAS database, at this time.

For more information on backing up databases, refer to the *DB2 Administration Guide*. For more information on the syntax of the backup command, refer to the *DB2 Command Reference*.

Oracle8 databases and tablespaces

Back up your existing Oracle8® tablespaces and datafiles for Commerce Suite 5.1 in accordance with the product documentation from Oracle® (for example, o816 and was).

A typical way to back up and restore user data in Oracle8i® is by using the export and import utilities provided by Oracle8i. By using the export and import utilities, the entire database can be backed up , a schema, or even a single table. You should run this utility from the Oracle server machine as these utilities are not available on the Oracle8i client.

Backing up a complete Oracle database

To back up a complete Oracle database you can use the export utility as follows:

1. Open a command prompt window.

2. Switch to the oracle user ID.

```
su - oracle
```

3. Enter the following command to run the export utility:

```
exp dba_user/password@service_name full=y file=file_name.dmp
   log=log_file.log
```

where:
- *dba_user* is the Oracle database administrator user ID, for example `oracle`.
- *password* is the associated password for *dba_user*.
- *service_name* is the Oracle service identifier (for example, `o816`).
- *file_name.dmp* is the file where the backup is stored.
- *log_file* is the name of the log file where status messages are stored.

4. Check the log file after the export completes. Ensure that there are no errors present in the log or else you may not have valid backup of your database. If you have any errors, consult the Oracle8i product documentation to assist you in troubleshooting any problems you may encounter.

Restoring a complete Oracle database

To restore a complete Oracle database you can use the import utility as follows:

1. Open a command prompt window.
2. Switch to the `oracle` user ID.

```
su - oracle
```

3. You must have a database before importing the data. Users will be created as part of the full import. Enter the following command to run the import utility:

```
imp dba_user/password@service_name full=y file=file_name.dmp
   log=log_file.log
```

where:
- *dba_user* is the Oracle database administrator user ID, for example `oracle`.
- *password* is the associated password for *dba_user*.
- *service_name* is the Oracle service identifier (for example, `o816`).
- *file_name*.dmp is the file where the backup is stored.
- *log_file*.log is the name of the log file where status messages are stored.

4. Check the log file after the import completes. Ensure that there are no errors present in the log or else you may not have restored the database properly. If you have any errors, consult the Oracle8i™ product documentation to assist you in troubleshooting any problems you may encounter.

Backing up a schema

To back up a schema, the schema owner can export his own data or a database administrator can export any schema or multiple schemas as follows:

1. Open a command prompt window.
2. Switch to the `oracle` user ID.

```
su - oracle
```

3. Export the schema.
   - A schema owner can export his own data as follows:

```
exp schema_owner/schema_owner_password@service_name file=file_name.dmp
   log=log_file.log
```

where:
   - *schema_owner* is the schema owner user ID.
   - *schema_owner_password* is the associated password for *schema_owner*.
   - *service_name* is the Oracle service identifier (for example, `o816`).

- *file_name*.dmp is the file where the backup is stored.
- *log_file*.log is the name of the log file where status messages are stored.
- An Oracle database administrator can export any schema, or multiple schemas at the same time:

```
exp dba_user/dba_user_password@service_name
  owner=schema_owner1,schema_owner2 file=file_name.dmp
  log=log_file.log
```

where:
  - *dba_user* is the database administrator user ID.
  - *dba_user_password* is the associated password for *dba_user*.
  - *service_name* is the Oracle service identifier (for example, o816).
  - *schema_owner1* and *schema_owner2* represent the schema owner user IDs.
  - *file_name*.dmp is the file where the backup is stored.
  - *log_file*.log is the name of the log file where status messages are stored.
4. Check the log file after the export completes. Ensure that there are no errors present in the log or else you may not have valid backup of your schema. If you have any errors, consult the Oracle8i product documentation to assist you in troubleshooting any problems you may encounter.

Restoring a schema

A schema owner can import his own data from the file to where the data was exported, and a database administrator can import any schema or multiple schemas at the same time:

1. Open a command prompt window.
2. Switch to the `oracle` user ID.

   `su - oracle`
3. Import the schema.
   - A schema owner can import his own data from the file to where the data was exported. A schema (user) should be created before running this command:

```
imp schema_owner/schema_owner_password@service_name
  fromuser=schema_owner full=n touser=schema_owner
  file=file_name.dmp log=log_file.log
```

where:
  - *schema_owner* is the schema owner user ID.
  - *schema_owner_password* is the associated password for *schema_owner*.
  - *service_name* is the Oracle service identifier (for example, o816).
  - *file_name*.dmp is the file where the backup is stored.
  - *log_file*.log is the name of the log file where status messages are stored.
   - An Oracle database administrator can export any schema, or multiple schemas at the same time. A DBA can also import the data of a schema to a different schema name:

```
imp dba_user/dba_user_password@service_name
  fromuser=schema_owner1,schema_owner2 touser=schema_owner1,schema_owner2
  file=file_name.dmp log=log_file.log full=n
```

where:
  - *dba_user* is the database administrator user ID.
  - *dba_user_password* is the associated password for *dba_user*.

- *service_name* is the Oracle service identifier (for example, o816).
- *schema_owner1* and *schema_owner2* represent the schema owner user IDs.
- *file_name*.dmp is the file where the backup is stored.
- *log_file*.log is the name of the log file where status messages are stored.

4. Check the log file after the export completes. Ensure that there are no errors present in the log or else you may not have restored the schema correctly. If you have any errors, consult the Oracle8i product documentation to assist you in troubleshooting any problems you may encounter.

Backing up a table name

To back up a table name you can use the export utility as follows:

1. Open a command prompt window.
2. Switch to the `oracle` user ID.

   ```
   su - oracle
   ```

3. Export the table name.
   - A schema owner can export one or more of his own tables as follows:

     ```
     exp schema_owner/schema_owner_password@service_name
       tables=table_name file=file_name.dmp log=log_file.log
     ```

     where:
     - *schema_owner* is the schema owner user ID.
     - *schema_owner_password* is the associated password for *schema_owner*.
     - *service_name* is the Oracle service identifier (for example, o816).
     - *table_name* is the table name that is exported (for example, USERS or KEYS).
     - *file_name*.dmp is the file where the backup is stored.
     - *log_file*.log is the name of the log file where status messages are stored.
   - An Oracle database administrator can export any table in any schema, or multiple tables at the same time:

     ```
     exp dba_user/dba_user_password@service_name
       tables=schema_owner.table_name
       file=file_name.dmp log=log_file.log
     ```

     where:
     - *dba_user* is the database administrator user ID.
     - *dba_user_password* is the associated password for *dba_user*.
     - *service_name* is the Oracle service identifier (for example, o816).
     - *schema_owner* is schema owner user ID for the table being exported (for example, MSER or EJSADMIN).
     - *table_name* is the table name that is exported (for example, USERS or SERVER_TABLE).
     - *file_name*.dmp is the file where the backup is stored.
     - *log_file*.log is the name of the log file where status messages are stored.

4. Check the log file after the import completes. Ensure that there are no errors present in the log or else you may not have restored the schema correctly. If you have any errors, consult the Oracle8i product documentation to assist you in troubleshooting any problems you may encounter.

Restoring a table name

To restore a table name you can use the import utility as follows:

1. Open a command prompt window.

2. Switch to the `oracle` user ID.

   `su - oracle`

3. Import the table name.

   - A schema owner can import one or more of his own tables from the file to where the data was exported:

     ```
     imp schema_owner/schema_owner_password@service_name
        tables=table_name full=n file=file_name.dmp
        log=log_file.log
     ```

     where:
     - *schema_owner* is the schema owner user ID.
     - *schema_owner_password* is the associated password for *schema_owner*.
     - *service_name* is the Oracle service identifier (for example, `o816`).
     - *table_name* is the table name that is exported (for example, `USERS` or `KEYS`).
     - *file_name*`.dmp` is the file where the backup is stored.
     - *log_file*`.log` is the name of the log file where status messages are stored.

   - An Oracle database administrator can import any table in any schema or multiple tables at the same time:

     ```
     imp dba_user/dba_user_password@service_name
        tables=schema_owner.table_name
        file=file_name.dmp log=log_file.log
     ```

     where:
     - *dba_user* is the database administrator user ID.
     - *dba_user_password* is the associated password for *dba_user*.
     - *service_name* is the Oracle service identifier (for example, `o816`).
     - *schema_owner* is schema owner user ID for the table being exported (for example, `MSER` or `EJSADMIN`).
     - *table_name* is the table name that is exported (for example, `USERS` or `SERVER_TABLE`).
     - *file_name*`.dmp` is the file where the backup is stored.
     - *log_file*`.log` is the name of the log file where status messages are stored.

4. Check the log file after the import completes. Ensure that there are no errors present in the log or else you may not have restored the table correctly. If you have any errors, consult the Oracle8i product documentation to assist you in troubleshooting any problems you may encounter.

Chapter 3. Upgrading your software

This chapter describes how to upgrade your software to the levels required by WebSphere Commerce 5.4. Prior to upgrading your software, you should backup your Commerce Suite 5.1 system, including the database. See Chapter 2, "Backing up Commerce Suite 5.1," on page 27 on how to perform a system backup.

If you need to stop any of your Commerce Suite 5.1 software components, see Appendix A, "Starting and stopping WebSphere Commerce Suite 5.1 components," on page 153.

> **Important**
>
> The information in this chapter describes how to upgrade your software for a single-tier environment only. If you have are using a remote web server (3–tier environment) you need to refer to the *WebSphere Commerce 5.4 Installation Guide* sections describing the installation and setup procedures for a remote web server and remote database.

Mapping the WebSphere Commerce Suite 5.1 and WebSphere Commerce 5.4 IBM software

The following tables map the version levels and default installation paths between Commerce Suite 5.1 and WebSphere Commerce 5.4 for most of the software packaged with either Commerce Suite 5.1 or WebSphere Commerce 5.4.

Table 4. Supplied software versions and installation paths for WebSphere Commerce 5.4 for AIX

Software	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
WebSphere Commerce	Version 5.1	Version 5.4
	/usr/lpp/CommerceSuite	/usr/WebSphere/CommerceServer
DB2 Universal Database™ Enterprise Edition	Version 7.1.0.20 for English versions and Version 7.1.0.24 for National Language versions	Version 7.1.0.55 (DB2 7.2 FixPak 5)
	/usr/lpp/db2_07_01	/usr/lpp/db2_07_01
IBM WebSphere Application Server — Advanced Edition	Version 3.5.2 (Version 3.5 plus FixPak 2 and eFixes)	Version 4.0.2 (Version 4.0.1 plus Version 4.0.2 PTF and eFixes)
	/usr/IBMWebAS	/usr/WebSphere/AppServer
IBM HTTP Server	Version 1.3.12	Version 1.3.19.1
	/usr/HTTPServer	/usr/HTTPServer
IBM SDK for Java	Version 1.2.2	Version 1.3.1
	/usr/jdk_base	/usr/WebSphere/AppServer/java
DB2 UDB Text Extender	Version 7.1	Version 7.1
	/usr/lpp/db2tx_07_01	/usr/lpp/db2tx_07_01
IBM WebSphere Payment Manager	Version 2.2	Version 3.1.2
	/usr/lpp/PaymentManager	/usr/lpp/PaymentManager

Table 4. Supplied software versions and installation paths for WebSphere Commerce 5.4 for AIX (continued)

Software	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
IBM SecureWay® Directory Server	Version 3.2.1	Version 3.2.1
Blaze Advisor Rule Server	Version 3.1.2	Version 3.1.3
	/usr/lpp/CommerceSuite/ blaze	/usr/WebSphere/CommerceServer/ blaze
Macromedia LikeMinds Personalization Server	Version 5.15.2.1	Version 5.2.x

Upgrading your hardware

If your current machine does not meet any of the following hardware requirements, you need to make any required upgrades to the hardware in order to meet the requirements.

- You require a dedicated machine from either the IBM @server™ pSeries™ or IBM RS/6000® family of machines such as the following:
  - RS64: @server pSeries 620/660/680, RS/6000 F80/H80/M80
  - Power3: @server pSeries 640, RS/6000 44P-170 or 44P-270
  - Power4: Regatta

  You must also meet the following hardware requirements
  - A 375 MHz processor.
  - A minimum of 768 MB of random access memory (RAM) per processor.
  - An additional 512 MB of RAM per WebSphere Commerce instance.
  - A minimum of 6 GB of free disk space, with the following recommended allocated file sizes:
    - /usr 4 GB
    - /tmp 1 GB
    - /home 1 GB
  - Enough free disk space on your machine to accommodate back ups of the Commerce Suite 5.1 database, in addition to the above requirement. This additional disk space should be at least three times the size of your current WebSphere Commerce Suite 5.1 database, as the database migration script will perform three back ups of the database during migration.
  - A minimum of 1 GB of paging space per processor.
  - A CD-ROM drive.
  - A graphics capable monitor.
- A local area network (LAN) adapter that supports the TCP/IP protocol.

For more information, see the preinstallation section of the *WebSphere Commerce 5.4 Installation Guide*.

Upgrading your operating system

WebSphere Commerce 5.4 supports AIX version 5.1.0. If you are on version 4.3.3 of AIX, you need to upgrade your operating system level to version 5.1. The following is a list of minimum system requirements for AIX 5.1:

- 64 MB physical memory

- 128 MB disk paging space
- 536 Mb disk page space for the base AIX install image

The 32- and 64-bit kernel have same requirements. Machines that work with AIX 5.1 are:
- RS64 -- pSeries 620/660/680; RS/6000 F80/H80/M80
- Power3 -- pSeries 640; RS/6000 44P-170 & -270
- Power4 Regatta

For more information, on operating system requirements including any maintenance levels required, see the *WebSphere Commerce 5.4 Installation Guide*. For information on migrating your AIX operating system from version 4.3.3 to version 5.1, see the migration section of the *AIX 5L™ Version 5.1 Installation Guide* available at following web site:

`http://www.ibm.com/servers/aix/library/techpubs.html`

- Check your OS level by issuing this command: `oslevel`
- The following filesets must be installed in your AIX 5.1.0.0 installation:
  - `X11.adt.lib`
  - `X11.base.lib`
  - `X11.base.rte`
  - `X11.motif.lib`
  - `bos.adt.base`
  - `bos.adt.include`
  - `bos.rte.net`
  - `bos.rte.libc`
  - `bos.net.tcp.client`

  If you are using Oracle as your database, you must also have the following filesets installed:
  - `bos.adt.lib`
  - `bos.adt.libm`

  For more information on determining if all the necessary filesets are installed and how to install them, refer to the *WebSphere Commerce 5.4 Installation Guide*.

Pre-install requirements check

A script is available on the WebSphere Commerce Disk 2 CD that will check your system to determine if you have the correct operating system, prerequisite software and pre-installed software. You should run this script before starting your installation to determine if your system meets all the needs of WebSphere Commerce. You can also run this script at various points during your installation to ensure that the software packages have been installed properly.

To run this script, perform the following steps:

1. Log on as user ID `root`.
2. If necessary, mount the WebSphere Commerce Disk 2 CD. To do this, type

   `mount CDROM_dir`

   where *CDROM_dir* is the directory that you want to specify as your mounted CD. For instructions on allocating a CD file system, see the section "Allocate a CD file system" in the *WebSphere Commerce 5.4 Installation Guide*.

3. Copy the `./wc54aixpreq.sh` from the `Software_Patches` directory on the CD to a temporary directory on your machine.
4. Form the temporary directory, run the script by typing:

   `./wc54aixpreq.sh`

Upgrading non-IBM software

This section describes how to upgrade non-IBM software components that are supported by Commerce Suite. This includes:

- Oracle database management system
- Domino™ Web Server
- Web browser

Upgrading Oracle databases

If you are using Oracle 8.1.6 with Commerce Suite 5.1, you need to upgrade your version of Oracle to Oracle 8.1.7 and apply PatchSet 2. To upgrade your version of Oracle to version 8.1.7, consult the *Oracle8i Migration Release 3 (8.1.7)* document — part number A86632–01. You can access PatchSet 2 from the following FTP site:

`ftp://oracle-ftp.oracle.com/server/patchsets/`

After you upgrade your version of Oracle version to version 8.1.7, ensure that the `COMPATIBLE` initialization parameter is set to 8.1.0. Otherwise you will receive errors when creating new tables with a `CLOB` field (for example, the `SCHERRLOG` table). For more information on this parameter, see the chapter on "Compatibility and Interoperability" in the *Oracle8i Migration Release 3 (8.1.7)* document .

In addition, update your Oracle initialization parameter so that you will be using the date format compatible with WebSphere Commerce 5.4:

1. Open the `$ORACLE_HOME/admin/wcs_database_name/pfile/initwcs_database_name.ora` file.
2. Add the following line into the file:

   `"nls_data_format = "DD/MM/YYYY H24:MI:SS.SSSSS"`
3. In addition, change `compatible = 8.0.5` to `compatible = 8.1.0` in the `initwcs_database_name.ora` file or you will not be able to display the "Publish Summary" page when publishing your store.
4. Shutdown and then restart the Oracle system.

Verify the modification by running the following SQL statement:

`select sysdate from dual;`

Netscape iPlanet

Netscape iPlanet does not run on AIX Version 5.0 and is thus unsupported by WebSphere Commerce 5.4. If you are using Netscape iPlanet in WebSphere Commerce 5.1, it is recommended that you switch to another supported Web server such as IBM HTTP Server while you are at WebSphere Commerce 5.1, prior to migrating to WebSphere Commerce 5.4.

Internet Explorer 5.5 or later

You can only access the WebSphere Commerce tools and online help using Microsoft® Internet Explorer 5.5 from a machine running a Windows operating system on the same network as your WebSphere Commerce machine. You must use Internet Explorer full version 5.50.4522.1800 (also known as Internet Explorer

5.5 Service Pack 1 and Internet Tools) or later with the latest critical security
updates from Microsoft— prior versions do not support full functionality of
WebSphere Commerce tools.

You can download Internet Explorer from Microsoft's Downloads Page:
`http://www.microsoft.com/downloads/`

Shoppers can access Web sites by using any of the following Web browsers, all of
which have been tested with WebSphere Commerce:
- Any version of Netscape Navigator supported with Netscape Communicator 4.6,
  including Netscape Navigator 4.04, and 4.5
- Netscape Navigator 3.0 and 4.0 or above for Macintosh
- Microsoft Internet Explorer 4 and 5
- AOL 5 and 6

Upgrading to WebSphere Application Server 4.0.1

Note: After you install WebSphere Commerce 5.4, you run the `wc54efixunix.sh`
script as described in "Upgrading your WebSphere Application Server
version" on page 60. This script will apply the WebSphere Application
Server 4.0.2 PTF and apply any eFixes required by WebSphere Commerce
5.4.

Before you start, be sure to read "Transition overview" on page 19 that describes
the reorganization of the Version 4.0 product relative to Version 3.x. A summary of
the product migration process follows. Most of this is done for you by
theWebSphere Application Server 4.0.1 installation and migration program.

1. Back up the current administrative configuration as described in "Backing up
   the current configuration" on page 40.
2. Remove the **WebSphere Commerce Suite** application server from the
   WebSphere Application Server administration console prior to upgrading to
   WebSphere Application Server 4.0.1 as described in "Removing the Commerce
   Server application server" on page 40. After installation of WebSphere
   Commerce 5.4, a new WebSphere Commerce application server will be created.
3. Follow the steps in "Using the WebSphere Application Server automated
   migration support" on page 41 to backup the remaining IBM WebSphere
   Application Server 3.5 administrative configurations and user data files, and to
   set up the migration environment. Note that this backup does not include the
   configuration for the WebSphere Commerce application server. Use the default
   `was3_backup` or `was_migration` backup directories. There will be an initial
   premigration phase to perform the backups when you first run the installation
   and migration program.
4. Stop and uninstall the current version of IBM HTTP Server and IBM
   WebSphere Application Server as described in "Uninstalling IBM HTTP Server"
   on page 42 and "Uninstalling IBM WebSphere Application Server" on page 42.
5. Drop the administrative database used for the previous installation of
   WebSphere Application Server as described in "Dropping the WebSphere
   Application Server Repository" on page 43.
6. Upgrade DB2 as described in "Upgrading to DB2 7.1 FixPak 3 (Version
   7.1.0.41)" on page 44 and "Upgrading to DB2 7.2 FixPak 5 (DB2 7.1.0.55)" on
   page 47.

7. Install the new version of IBM WebSphere Application Server as described in "WebSphere Application Server 4.0.1 installation and postinstallation phase" on page 50. This includes upgrading any prerequisites such as the level of DB2.

8. Start the new administrative server.

9. Restore the configuration in the new installation using the `was3_backup` directory.

Backing up the current configuration

Before you begin the transition to WebSphere Application Server 4.0.1, you need to back up your existing WebSphere Application Server configurations:

1. Ensure that the WebSphere Application Server administrative server is running.

2. Create a backup directory where you can export the configurations (for example, `/was_backup_all`).

3. Back up your current administrative configurations using the WebSphere Application Server XMLConfig utility with the `-export` flag. From a command line, switch to the `/usr/IBMWebAS/bin` path and issue the following command:

   `./XMLConfig.sh -export back_up_file_name.xml -adminNodeName your_host_name`

   where *back_up_file_name*.xml is the fully qualified name of your back up file for your current WebSphere Application Server configurations. For example:

   `./XMLConfig.sh -export /was_backup_all/was35_backup_all.xml -adminNodeName montreal`

All items that are exported are displayed on the screen.

Removing the Commerce Server application server

You need to remove the WebSphere Commerce application server from the WebSphere Application Server administration console as follows:

1. Start the WebSphere Application Server if it is stopped.

2. In the WebSphere Application Server Administrative Console, do the following:
   a. Expand the node for your host name.
   b. Right-click the WebSphere Commerce Suite Application Server that corresponds to your instance and select **Stop**.
   c. When the application server for your instance has stopped, right-click the instance again and select **Remove**.
   d. Remove the WebSphere Commerce Suite data source by right-clicking the data source entry and selecting **Remove**.
   e. Right-click on the WebSphere Commerce Suite JDBC Driver and select **Uninstall**. In the window that displays, select your host name and click **Uninstall**.
   f. Remove the defined JDBC driver by right-clicking the JDBC driver and selecting **Remove**.

From the WebSphere Application Server 3.5.4 Administrative Console, ensure that the following entries have been removed from WebSphere Application Server:

- `WebSphere Commerce Application Server`
- `WebSphere Commerce DataSource`
- `WebSphere Commerce JDBC Driver`
- Virtual host — `VH_instance_name_tools` (for example, `VH_demo_tools`

Using the WebSphere Application Server automated migration support

In IBM WebSphere Application Server Version 4.0, automated migration support is part of the installation program. In the preinstallation phase, the installation program detects previously installed versions, collects information about how you want the migrated installation to look, and exports the current administrative configuration. After you have updated product prerequisites, you run the second phase, which installs the new version of the product and imports the backed-up administrative configuration.

Note: Before you begin, make sure the WebSphere Application Server administrative server and your Web server are running.

Preinstallation phase

In the preinstallation phase:

1. When you start the installation program, it automatically detects previously installed versions of the product and displays them in a list. If the installation program supports migration from a selected version, a **Perform migration** check box appears above the list.

2. The installation program prompts you for the following information:
   - Backup directory.

     Override the default directory (`/usr/IBMWebAS/migration/backup`) with another directory such as `/was3_backup`, since the default directory gets deleted in the following steps.
   - Directory for temporary staging.

     Override the default directory (`/usr/IBMWebAS/migration/tmp`) with another directory such as `/was3_migration`, since the default directory gets deleted in the following steps.
   - Directory for migration log.

     Override the default directory (`/usr/IBMWebAS/migration/logs`) with another directory such as `/was_migration/logs`, since the default directory gets deleted in the following steps.

3. At the confirmation prompt to begin the migration process, click **Yes** to continue.

4. If you have not started the WebSphere Application Server administrative server, the installation program will prompt you to start it.

5. Click **Start Migration**. The installation program exports the current configuration and displays the migration log file. The migration log is displayed only after migration is finished.

6. When the installation program prompts you to check the migration log file, do one of the following:
   - If the migration log file indicates success, click **Finished**.

     The next time you run the installation program in "WebSphere Application Server 4.0.1 installation and postinstallation phase" on page 50, the next phase of the migration (installation and postmigration) will begin.
   - If the migration log file does not indicate success, click **Cancel** and correct the logged errors.

     In this case, the next time you run the installation program, the migration will start from the beginning.

Note: If the migration log file indicates problems with the migration and you click
Finished, you cannot rerun this phase of migration until you delete the file
`/tmp/WAS_Migration_temp.properties`. When this phase finishes successfully,
do the following:

1. Migrate prerequisites.
2. If you are installing Version 4.0 into the same directory structure as
   Version 3.x, do one of the following:
   - If the migration backup directory is within the Version 3.x directory
     structure, retain the migration backup directory but delete the rest of
     the Version 3.x directory structure.
   - Otherwise, delete the entire Version 3.x directory structure.
3. Restart the installation program.

Uninstalling IBM HTTP Server and WebSphere Application Server

After you complete the premigration phase, you need to uninstall the previous
versions of IBM HTTP Server and WebSphere Application Server. For DB2
databases, if you have no further requirement for the previous `WAS` repository, you
can drop it as you create a new `WAS40` repository after you install WebSphere
Application Server 4.0.1. You also need to upgrade DB2 to the correct level before
you install WebSphere Application Server 4.0.1.

Uninstalling IBM HTTP Server:

Notes:

1. If you uninstall IBM HTTP Server, you must also uninstall WebSphere
   Application Server.
2. If you reinstall IBM HTTP Server and WebSphere Application Server, you must
   install IBM HTTP Server *before* reinstalling WebSphere Application Server since
   WebSphere Application Server modifies your Web server's configuration file.

To uninstall IBM HTTP Server, do the following:

1. Stop IBM HTTP Server, as described in "Starting and stopping IBM HTTP
   Server" on page 154.
2. If you have created or customized any files in the /usr/HTTPServer directory
   or its subdirectories, and you wish to retain them, back them up to a directory
   that is not used by any WebSphere Commerce Suite component.
3. If you want to save a copy of `/usr/HTTPServer/conf`, copy it to a different
   name.
4. While logged on to user ID `root`, type `smitty` and remove all `gskit.*` and all
   `http_server.*` components.
5. Uninstall the packages.
6. Remove any remaining IBM HTTP Server files and directories as follows:
   a. If the `/usr/HTTPServer` directory exists, remove it by typing the following:
      ```
      cd /usr
       rm -r HTTPServer
      ```

You must uninstall WebSphere Application Server immediately.

Uninstalling IBM WebSphere Application Server: To uninstall your previous
version of WebSphere Application Server, do the following:

1. Stop any Commerce Suite instances that may be started in the WebSphere
   Application Server Administrative Console.
2. Close the WebSphere Application Server Administrative Console.

3. Stop WebSphere Application Server, as described in "Starting and stopping WebSphere Application Server" on page 153.

4. If you have created or customized any files in the /usr/IBMWebAS directory or its subdirectories, and you wish to retain them, back them up to a different directory.

5. Run SMITTY.

6. Select the WebSphere filesets. The filesets names start with IBMWebAS.

7. Uninstall the packages.

8. If the /usr/IBMWebAS directory still exists, remove it by typing the following:

```
cd /usr
rm -r IBMWebAS
```

Dropping the WebSphere Application Server Repository: If you are using DB2, and you no longer require the WebSphere Application Server administrative database used for the previous installation (for example, the WAS database) you can drop the database as follows:

1. Back up the database. You can use the database management system or simply rename the database. Refer to "Backing up your databases" on page 28 for details on backing up your database.

2. Remove your WebSphere Application Server database by typing the following in a command window:

```
db2 drop db db_name
```

Note that it is not mandatory to drop the IBM WebSphere Application Server 3.5.x administrative database since the WebSphere Application Server 4.0.x repository database which gets created later is named differently than the IBM WebSphere Application Server 3.5.x administrative database.

If you are using Oracle, you can either create a new Oracle user for the WebSphere Application Server 4.0.x repository, or you can continue to use the same Oracle user as you used for IBM WebSphere Application Server 3.5.x.

If you wish to create create a new user for WebSphere Application Server 4.0.x:

1. Create a new Oracle database user ID:

```
CREATE USER user_id IDENTIFIED BY user_password DEFAULT
  TABLESPACE table_space_name QUOTA UNLIMITED ON table_space_name;
```

where *user_id* is the new user's name that you create, *user_password* is the new user's password, and *table_space_name* is the tablespace name on which *user_id* is based.

2. Grant DBA authentication to the new user ID:

```
GRANT DBA TO user_id;
```

where *user_id* is the new user's name that you have created.

If you wish to use the same user as before (EJSADMIN) for WebSphere Application Server 4.0.x:

1. Drop the EJSADMIN Oracle database user:

```
DROP USER EJSADMIN CASCADE;
```

2. Recreate the EJSADMIN Oracle database user:

```
CREATE USER EJSADMIN IDENTIFIED BY EJSADMIN DEFAULT
  TABLESPACE table_space_name QUOTA UNLIMITED ON table_space_name;
```

where *table_space_name* is your tablespace name on which user EJSADMIN is based.

3. Grant DBA authentication to the EJSADMIN user:

   ```
   GRANT DBA TO EJSADMIN;
   ```

4. Assign TEMP tablespace to EJSADMIN:

   ```
   ALTER USER EJSADMIN TEMPORARY TABLESPACE TEMP;
   ```

Upgrading to DB2 7.2 FixPak 5

This section describes how to upgrade your level of DB2 to that required by WebSphere Commerce 5.4

Upgrading to DB2 7.1 FixPak 3 (Version 7.1.0.41): To upgrade to DB2 7.2 FixPak 5 from DB2 7.1 FixPak 2, you must first apply DB2 7.1 FixPak 3 in order to upgrade your DB2 level to Version 7.2.

DB2 7.1 FixPak 3 is provided for your convenience in the WebSphere Commerce Disk 3 CD.

The English version of FixPak 3 is also available for download from the following DB2 support web site:

```
ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2aixv7/FP3_U475375/
```

For the national language version of FixPak 3, see the following Web site:

```
http://www14.software.ibm.com/webapp/download/postconfig.jsp?id=4000604
 &pf;=Platform+Independent&v;=Version+7&e;=&cat;=&s;=p
```

Prerequisites: You should have DB2 UDB V7.1 already installed before installing this FixPak.

To determine if the required prerequisites are installed, issue the following command:

```
lslpp -al "db2_07_01*"
```

The output of the command should include the following:

```
db2_07_01.client      7.1.0.x
```

Prior to installation: Logon as `root` and enter the following commands for each instance:

```
su - ioname
. $HOME/sqllib/db2profile
db2 force applications all
db2 terminate
db2stop
db2licd end     # run at each physical node
exit
```

where *ioname* is the instance owner name (for example, db2inst1).

Then, while still logged on as root, enter the following commands:

```
su - aname
. $HOME/sqllib/db2profile
db2admin stop
exit
/usr/sbin/slibclean
```

where *aname* is the DB2 administration server name.

Installing DB2 7.2 FixPak 3: Mount the WebSphere Commerce Disk 3 CD on your machine and switch to the root directory of the CD.

Install the selected updates using either SMIT or `installp`. There are two steps to installing this FixPak. The first step is to install the `.bff` files. The second step is to install the filesets in the `extras` subdirectory. The `extras` subdirectory is located in the same directory as the `.bff` files. To list the installable options in this FixPak, use:

```
installp -ld device
```

where *device* is the name of the device where the updated image resides.

To install with all the prerequisites in apply state, select **Software Installation & Maintenance --> Update Installed Software to Latest Level (Update ALL)** in SMIT. To skip directly to this menu, use the following command:

```
smit update_all
```

If you would like to be able to reject (uninstall) the FixPak afterwards, make sure to select the following options in SMIT:

```
COMMIT software updates = NO
SAVE replaced files = YES
```

You can later commit or reject the FixPak using SMIT again.

You can later commit or reject the FixPak using SMIT again.

This FixPak has an additional subdirectory in the installation image. The name of this subdirectory is `extras`. This `extras` subdirectory contains the filesets for the double-byte and unicode conversion tables, specifically, the following filesets:

```
db2_07_01.cnvucs
db2_07_01.conv.jp
db2_07_01.conv.kr
db2_07_01.conv.sch
db2_07_01.conv.tch
```

To install the filesets in the extras subdirectory, select the following menu options in SMIT:

```
smit install_all
```

Note: Installing the `extras` subdirectory will automatically commit the above filesets. You will not be able to reject the FixPak if you install the `extras` subdirectory.

Since future FixPaks will require this FixPak to be already installed, they will no longer include the ″extras″ subdirectory. This means that you must have these filesets installed before going on to future FixPaks.

For more information about the installp command or SMIT, refer to the online documentation by issuing the `man installp` or `man smit` commands.

Post installation: There are a number of items you need to perform as follows after installation of the FixPak such as rebinding bind files as described below. For additional considerations, see the DB2 7.2 FixPak 3 Readme document.

Updating instances:

Note: You need to be logged on as root to update the instances.

For each instance, issue the command:

```
/usr/lpp/db2_07_01/instance/db2iupdt iname
```

where *iname* is the instance name (for example, db2inst1).

Note: If you are running both Enterprise - Extended Edition (EEE) and Enterprise Edition (EE) on the same machine, then you need to run db2iupdt with the "-k" option. Otherwise, it will check to see if EEE filesets exist; if they do, it will perform the update as if it was an EEE instance.

For the DAS instance, issue the command:

```
/usr/lpp/db2_07_01/instance/dasiupdt dasname
```

where *dasname* is the DAS instance name. For more information on the db2iupdt or dasiupdtcommands, run db2iupdt or dasiupdt with the "-h" option.

Rebinding bind files: Due to the new bind (.bnd) files that have been shipped with this FixPak, you will need to rebind your DB2 utilities against all your databases after the FixPak installation. This step is necessary for the fixes in this FixPak to become effective. The procedure of rebinding, outlined below, only needs to be performed once per database.

Note: The Run-Time Client cannot be used to bind the database utilities (import, export, reorg, the Command Line Processor) and DB2 CLI bind files to each database before they can be used with that database. You must use the DB2 Administration Client or the DB2 Application Development Client instead.

Rebinding to non-DRDA DB2 databases: After applying fixes, you must rebind the DB2 utilities as follows from a command prompt:

```
db2 terminate
db2 CONNECT TO dbname


db2 BIND path/@db2ubind.lst GRANT PUBLIC
db2 BIND path/@db2cli.lst GRANT PUBLIC
db2 terminate
```

where *dbname* is the name of a database to which the utilities should be bound. *path* is the full path name of the directory where the bind files are located, such as /sqllib/bnd. DB2UBIND.LST and DB2CLI.LST contain lists of commonly required bind files used for DB2.

Rebinding specific packages: If you want to bind only the specific .bnd files to the database, issue the following commands for each database from a command prompt: :

```
db2 terminate
db2 CONNECT TO dbname

db2 BIND path/bind_file BLOCKING ALL GRANT PUBLIC
db2 terminate
```

where *bind_file* is the name of the bind file, and *dbname* is the name of your database.*path* is the full path name of the directory where your bind files are located.

Upgrading to DB2 7.2 FixPak 5 (DB2 7.1.0.55):   To upgrade to DB2 7.2 FixPak 5 from DB2 7.1 FixPak 2, you must first apply DB2 7.1 FixPak 3 as described in the previous section, before applying DB2 7.2 FixPak 5.

DB2 7.2 FixPak 5 is provided for your convenience in the /Software_Patches/DB2_fp5 directory in the WebSphere Commerce Disk 2 CD.

The English version of FixPak 5 is also available for download from the following DB2 support web sites:

```
ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2aixv7/FP5_U480359/
```

For the national language version of the FixPak, see the following Web site:

```
http://www14.software.ibm.com/webapp/download/postconfig.jsp?id=4000604
 &pf=Platform+Independent&v=Version+7&e=&cat=&s=p
```

Prerequisites: You should have DB2 UDB V7.2 or DB2 UDB V7.1 with FixPak 3 or greater already installed before installing this FixPak.

To determine if the required prerequisites are installed, issue the following command:

```
lslpp -al "db2_07_01*"
```

The output of the command should include the following:

```
db2_07_01.client      7.1.0.40 or higher
```

Prior to installation: Logon as root and enter the following commands for each instance:

```
su - ioname
. $HOME/sqllib/db2profile
db2 force applications all
db2 terminate
db2stop
db2licd end      # run at each physical node
exit
```

where *ioname* is the instance owner name (for example, db2inst1).

Then, while still logged on as root, enter the following commands:

```
su - aname
. $HOME/sqllib/db2profile
db2admin stop
exit
/usr/sbin/slibclean
```

where *aname* is the DB2 administration server name.

Installing DB2 7.2 FixPak 5:   Mount the WebSphere Commerce Disk 2 CD on your machine and switch to the /Software_Patches/DB2_fp5 directory.

Install the selected updates using either SMIT or installp. To list the installable options in this FixPak, use:

```
 installp -ld device
```

where *device* is the name of the device where the updated image resides.

To install with all the prerequisites in apply state, select **Software Installation &
Maintenance --> Update Installed Software to Latest Level (Update ALL)** in
SMIT. To skip directly to this menu, use the following command:

```
smit update_all
```

If you would like to be able to reject (uninstall) the FixPak afterwards, make sure
to select the following options in SMIT:

```
COMMIT software updates = NO
SAVE replaced files = YES
```

You can later commit or reject the FixPak using SMIT again.

For more information about the installp command or SMIT, refer to the online
documentation by issuing the man installp or man smit commands.

Post installation: There are a number of items you need to perform as follows after
installation of the FixPak such as rebinding bind files as described below. For
additional considerations, see the DB2 7.2 FixPak 5 Readme document.

Updating instances:

Note: You need to be logged on as root to update the instances.

For each instance, issue the command:

```
/usr/lpp/db2_07_01/instance/db2iupdt iname
```

where *iname* is the instance name (for example, db2inst1).

Note: If you are running both Enterprise - Extended Edition (EEE) and Enterprise
Edition (EE) on the same machine, then you need to run db2iupdt with the
"-k" option. Otherwise, it will check to see if EEE filesets exist; if they do, it
will perform the update as if it was an EEE instance.

For the DAS instance, issue the command:

```
/usr/lpp/db2_07_01/instance/dasiupdt dasname
```

where *dasname* is the DAS instance name. For more information on the db2iupdt or
dasiupdtcommands, run db2iupdt or dasiupdt with the "-h" option.

Rebinding bind file: Due to the new bind (.bnd) files that have been shipped with
this FixPak, you will need to rebind your DB2 utilities against all your databases
after the FixPak installation. This step is necessary for the fixes in this FixPak to
become effective. The procedure of rebinding, outlined below, only needs to be
performed once per database.

Note: The Run-Time Client cannot be used to bind the database utilities (import,
export, reorg, the Command Line Processor) and DB2 CLI bind files to each
database before they can be used with that database. You must use the DB2
Administration Client or the DB2 Application Development Client instead.

Rebinding to non-DRDA DB2 databases: After applying fixes, you must rebind the
DB2 utilities as follows from a command prompt:

```
db2 terminate
db2 CONNECT TO dbname
```

```
db2 BIND path/@db2ubind.lst GRANT PUBLIC
db2 BIND path/@db2cli.lst GRANT PUBLIC
db2 terminate
```

where *dbname* is the name of a database to which the utilities should be bound.
path is the full path name of the directory where the bind files are located, such as
/sqllib/bnd. DB2UBIND.LST and DB2CLI.LST contain lists of commonly required
bind files used for DB2.

Rebinding specific packages: If you want to bind only the specific .bnd files to the
database, issue the following commands for each database from a command
prompt: :

```
db2 terminate
db2 CONNECT TO dbname

db2 BIND path/bind_file BLOCKING ALL GRANT PUBLIC
db2 terminate
```

where *bind_file* is the name of the bind file, and *dbname* is the name of your
database.*path* is the full path name of the directory where your bind files are
located.

Installing the DB2 Application Development Client: If your previous installation
of DB2 did not include the DB2 Application Development Client component, you
will need to install this component at the DB2 7.2 FixPak 5 level before upgrading
to WebSphere Application Server 4.0.1.

Install this component using the fully installable DB2 7.2 FixPak 5 CD provided
with WebSphere Commerce 5.4. For information on installing this component, see
the chapter on installing DB2 in the *WebSphere Commerce 5.4 Installation Guide*.

Configuring the database manager to use jdbc2 and shared memory: Before
creating a database with DB2, you must complete the following steps on your DB2
server machine:

1. Change to the db2inst1 user:

   ```
   su - db2inst1
   ```

2. Add the following lines to the bottom of the db2inst1 .profile:

   ```
   . ./sqllib/java12/usejdbc2
   EXTSHM=ON
   export EXTSHM
   db2set DB2ENVLIST=EXTSHM
   ```

3. Save the file.
4. Type db2stop
5. Type db2start
6. Type exit

Creating a database for WebSphere Application Server: Perform the following
steps to create a database named was40 and set the DB2 application heap size:

1. Ensure that you are logged in as the DB2 instance owner, db2inst1.
2. Ensure that DB2 is running.
3. Create a database named was40 by using the following command:

   ```
   db2 create database was40
   ```

   This process can take several minutes to complete.

4. Set the application heap size, by using the following command:

```
db2 update db config for WAS40 using applheapsz 256
```

5. When the database creation has finished, stop and start DB2 so that your changes to take effect:

```
db2stop
db2start
```

If an application heap size of 256 does not work for your system, increase it to 512.

6. Determine the TCP/IP service name by typing the following:

```
db2 get dbm cfg |grep -i SVC
```

7. Catalog the WebSphere Application Server database as if it were a remote database by typing the following:

```
db2 catalog tcpip node node_name remote full_host_name server
   TCP/IP_service_name
db2 catalog database WAS40 as WASLOOP at node node_name
```

Where *node_name* is the name you are assigning to this node, and *full_host_name* is the fully qualified host name of your database server. These commands are presented on separate lines for readability only; ensure that you type each one on a single line.

8. Test the connection by typing:

```
db2 connect to WASLOOP user username using password
```

Note: Later, during WebSphere Application Server 4.0.1 installation, in the Database Name (Database SID) field, type the name of the database — WASLOOP.

Verifying the connection to the database: Perform the following steps to verify a connection to database named was40 :

1. Ensure that you are logged in as the DB2 instance owner, db2inst1.

2. Connect to the database named was40 by using the following command:

```
db2 connect to was40
```

Correct output resembles the following:

```
Database Connection Information

    Database server = DB2/6000 7.2.1
    SQL authorization ID = DB2INST1
    Local database alias = WAS40
```

3. To disconnect from a database and log out as the DB2 instance owner, type exit at the command prompt.

WebSphere Application Server 4.0.1 installation and postinstallation phase

Before you perform the following steps to install WebSphere Application Server, see the WebSphere Application Server Supported Hardware, Software, and APIs Web site at:

```
http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html
```

to learn which products and fix levels are supported for use with WebSphere Application Server on your operating system.

The steps that follow describe how to install a configuration of WebSphere Application Server Advanced Edition that uses:

- AIX 4.3.3 or AIX 5.1
- IBM Java 2 Software Developer's Kit (SDK) 1.3.0
- IBM HTTP Server 1.3.19
- DB2 Universal Database (UDB) 7.2 FixPak 5

Ensure that you meet the following prerequisites:

1. Ensure that you are logged into the machine with superuser (root) privileges.
2. If a preexisting Web server on your system is running, stop the Web server. If you plan to install IBM HTTP Server 1.3.19 as part of the WebSphere Application Server installation and a version of IBM HTTP Server prior to 1.3.19 is already installed on your system, you must uninstall it in order for the WebSphere Application Server installation program to successfully install IBM HTTP Server 1.3.19.
3. If another Web server is running on your system, stop the Web server.
4. Insert the WebSphere Application Server CD-ROM into the CD-ROM drive.
5. If necessary, use the `mkdir` command to create a mount point for the CD-ROM. The following command creates a mount point at the directory `/cdrom`; you can mount the CD-ROM at any location on the machine's local file system.

   ```
   mkdir /cdrom
   ```

   The commands in these steps assume the CD-ROM is mounted at `/cdrom`. If you mount the CD-ROM at a different location, use that location when issuing commands.
6. Mount the CD-ROM drive by entering the following command:

   ```
   mount -o ro -v cdrfs /dev/cdnumber /cdrom
   ```

   In this command, *number* is the CD-ROM number for your system, usually 0 (zero). Note that this command assumes that the CD-ROM is mounted at `/cdrom`.
7. Ensure that your `DISPLAY` and `TERM` environment variables are set properly.
8. If the supported Web server or database you plan to use with WebSphere is newer than the version currently required by WebSphere Application Server, you must update the `prereq.properties` file or disable the prerequisite checking functionality before installing WebSphere Application Server. To obtain an updated `prereq.properties` file, download the latest version from the WebSphere Application Server Tools Web site.

   ```
   http://www.ibm.com/software/webservers/appserv/tools.html
   ```

   Ensure that the updated `prereq.properties` file is downloaded or copied into the local `/tmp` directory.

   To disable prerequisite checking functionality, perform the following steps:

   a. Copy the `prereq.properties` file from the `/cdrom` directory to the `/tmp` directory on the machine on which you plan to install WebSphere Application Server.
   b. Open the `prereq.properties` file in a text editor and disable prerequisite checking for an individual component by changing the value of the specific key from 1 to 0.
   c. Save the edited `prereq.properties` file.

After you have met all product prerequisites for WebSphere Application Server 4.0.1, you can begin the next phase.

1. When you restart the WebSphere Application Server 4.0.1 installation program, it may detect the following situations:
   - Any remaining prerequisites that have not been upgraded appropriately. You must exit the installation program and ensure that these prerequisites are met.
   - A running application server, if the new installation directory is the same as the current one. You must stop WebSphere Application Server in order for the installation program to properly overlay the files.
2. The installation program takes you through the standard installation process as described below in "Installing WebSphere Application Server 4.0.1."

After the installation has completed:

- A postinstallation migration is performed. The installation program imports the configuration that was exported in the first phase and displays the migration log file.
- When you start the administrative server the first time, it will take longer to start up as it is loading the configuration file that was created during the migration process.

Installing WebSphere Application Server 4.0.1

These instructions assume the following:

- The machine has enough memory and disk space for your installation. See the WebSphere Application Server Supported Hardware, Software, and APIs Web site at:

  `http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html`

  for the requirements.
- If you plan to use IBM HTTP Server, you will select it for installation during the WebSphere Application Server installation process. If you plan to use a different supported Web server with WebSphere Application Server, you have already installed it on the same node as WebSphere Application Server.
- You have installed and configured a supported database for use with WebSphere Application Server.

Note: You must purchase and install any Web server except IBM HTTP Server before you install WebSphere Application Server. IBM HTTP Server is supplied with WebSphere Application Server and can be installed automatically during installation of WebSphere Application Server.

Perform the following steps to install WebSphere Application Server:

1. Navigate to the `/cdrom/aix` directory.
2. If you have not downloaded an updated `prereq.properties` file or disabled the prerequisite checking functionality, start the WebSphere Application Server installation program by using the `install.sh` command, as follows:

   `./install.sh`

   If you have downloaded an updated `prereq.properties` file or disabled the prerequisite checking functionality as detailed in step 8 on page 51, start the WebSphere Application Server installation program by using the `install.sh` command, as follows:

   `./install.sh -prereqfile /tmp/prereq.properties`

3. The Welcome to the IBM WebSphere Application Server Setup program dialog box opens. Click **Next** to continue.

4. The Install Options dialog box opens. Select **Custom Installation**, and click **Next**.

5. The Choose Application Server Components dialog box opens. Select the components you want to install and deselect the components you do not want to install. Note the following information:

   • The Java 2 Software Developer's Kit (SDK) is installed by default.

   • The Server, Admin, Samples, Application Assembly and Deployment Tools, IBM HTTP Server 1.3.19, and Web Server Plugins components are selected for installation by default.

   • If you plan to use WebSphere Application Server with IBM HTTP Server, ensure that the IBM HTTP Server 1.3.19 and Web Server Plugins options are selected.

   • If you plan to use WebSphere Application Server with a different supported Web server, ensure that the Web Server Plugins option is selected.

     Note: No plug-ins are required to launch the Application Server or the administrative console. However, for production applications, you will not be able to serve servlets without having installed a supported Web server and corresponding Web server plug-in.

     For non-production applications, you can use the internal HTTP transport system to serve servlets without installing a Web server plug-in by using the internal HTTP transport port 9080 . For example, to serve the sample snoop servlet by using the internal HTTP transport, enter the URL

     `http://local_host:9080/servlet/snoop`

     The internal HTTP transport mechanism is not designed for use in a production environment.

   • If you plan to install the Web server plug-in for IBM HTTP Server, you must select the `IBM HTTP Server 1.3.19.1` option, or have it already installed on the machine.

   • These instructions assume that you are installing all of the components.

   Click **Next** to continue.

6. If you selected the `Web Server Plugins` option, the Choose Application Server Components dialog box opens. Select the appropriate plug-in for your Web server, and click **Next**.

7. The Database Options dialog opens. Depending on the database you have installed, complete one of the following set of instructions:

   • **If you are using DB2**, perform the following steps in the Database Options dialog:

     a. In the Database Type field, select **DB2** from the pull-down menu.

     b. Ensure that **Remote DB** is not selected. For this example, the database and WebSphere Application Server are installed on the same node.

     c. In the Database Name (Database SID) field, type the name of the database,WASLOOP.

     d. In the DB Home field, type the full pathname of the home directory of the DB2 instance owner, `/home/db2inst1`, or specify the full pathname of the home directory by using the Browse button.

     e. The DB URL field cannot be edited.

f. The Server Name field cannot be edited.

g. The Port Number field cannot be edited.

h. In the Database User ID field, type the name of the database instance owner, `db2inst1`.

i. In the Database Password field, type the current password for the database instance owner.

j. Click **Next** to continue

- **If you are using Oracle**, perform the following steps in the Database Options dialog:

    a. In the Database Type field, select **Oracle** from the pull-down menu.

    b. Ensure that Remote DB is not selected. For this example, the database and WebSphere Application Server are installed on the same node.

    c. In the Database Name (Database SID) field, type the name of the Oracle database you created. For example, `ORA817.machine_name`.

    d. In the DB Home field, type the full pathname of the directory you created to contain the Oracle software and to be the home directory of the user named oracle, or specify the full path name of the directory by using the Browse button. This path should also be the value of the `ORACLE_HOME` environment variable.

    e. In the DB URL field, accept the default value:

      `jdbc:oracle:thin:@fully_qualified_domain_name:port_number:database_name`

      or specify a different URL for accessing the database.

    f. In the Server Name field, type the name of the machine on which the database is installed.

    g. In the Port Number field, type the port number used to access the database.

    h. In the Database User ID field, type the name of the database owner, `EJSADMIN`.

    i. In the Database Password field, type the current password for the database owner.

    j. Click **Next** to continue.

8. The Select Destination Directory dialog opens. Specify the directory in which you want to install WebSphere Application Server. You can either accept the default destination directory or specify a different one by typing the full pathname or by clicking Browse. Note that if you've selected IBM HTTP Server for installation, you cannot modify the destination directory. Click **Next** to continue.

9. The Install Options Selected dialog box opens. Verify that the information is correct and click **Install** to complete the installation.

10. Depending on the machine's configuration, the Location of Configuration files dialog box can open. It prompts you to enter the full pathname of the directory in which you want to store the specified Web server configuration file. Specify the full pathname of the file by typing it in the field or by clicking **Browse**.

11. The Setup Complete dialog box opens. To view the ReadMe file, ensure that **Yes, I want to view the ReadMe File** is selected and click **Finish**; the ReadMe file is displayed in a default browser window. To view the ReadMe file at a later time, deselect **Yes, I want to view the ReadMe File** and click **Finish** to exit from the WebSphere Application Server installation program.

12. The WebSphere Application Server - First Steps dialog box opens. You can use this GUI to access product information in the InfoCenter, start the administrative server, launch the administrative console, or launch the application assembly tool. Because you must first start and possibly configure the Web server before using WebSphere, close this dialog for now. You can launch the First Steps GUI at a later time by running the `firststeps.sh` script located in the `/usr/WebSphere/AppServer/bin` directory.

13. Unmount the CD-ROM before removing it from the CD-ROM drive by using the `umount` command, as follows:

    `umount /cdrom`

14. If you are using a Web server other than IBM HTTP Server, start the server. If you are using IBM HTTP Server and have installed it during the WebSphere Application Server installation, you may need to configure the Web Server to run it successfully.

    Perform the following steps to verify that IBM HTTP Server is installed and configured correctly:

    a. Ensure that the Web server is running. If not, start it by entering the following command:

       `/usr/HTTPServer/bin/apachectl start`

    b. Start a browser and enter the name of the local machine as the URL. If you see the IBM HTTP Server Web page, the server is installed and configured correctly.

    See the IBM HTTP Server documentation at the following Web site for more information about configuring IBM HTTP Server:

    `http://www.ibm.com/software/webservers/httpservers/library.html`

    To enable the Secure Sockets Layer (SSL) for IBM HTTP Server, see the IBM HTTP Server documentation for more information at the following Web site:

    `http://www.ibm.com/software/webservers/httpservers/doc/v1319/index.html`

15. If you are using DB2, proceed to "Testing the installation" on page 56. If you are using Oracle, proceed to "WebSphere Application Server configuration for Oracle users."

WebSphere Application Server configuration for Oracle users: If you are using an Oracle database, some manual configuration steps are required to ensure that WebSphere Application Server has full functionality within WebSphere Commerce. Before starting these steps you will need to know the fully qualified path of your Oracle home directory. For example, /oracle/u01/app/oracle/product/8.1.7.

To complete the manual configuration of WebSphere Application Server do the following:

1. Log in as `root`.

2. Change to the WebSphere Application Server `bin` directory by executing the following command in a terminal window:

   `cd /usr/WebSphere/AppServer/bin`

3. Copy `startupServer.sh` to `startupServer.sh.orig`.

4. Open `startupServer.sh` in a text editor.

5. Locate the following section of text:

   ```
   if [ "${DB_TYPE}" != "DB2" ]
   then
   {
   LD_LIBRARY_PATH=$WAS_HOME/bin:$WAS_HOME/lib:$DB_INSTANCE_HOME/lib:$LD_LIBRARY_PATH
   ```

```
LIBPATH=$WAS_HOME/bin:$WAS_HOME/lib:$DB_INSTANCE_HOME/lib:$LIBPATH
export LD_LIBRARY_PATH LIBPATH

  ${JAVA_EXE?} \
 -classpath $WAS_HOME/lib/bootstrap.jar:$CLASSPATH \
 -Dws.ext.dirs=$WAS_EXT_DIRS \
 -Djavax.rmi.CORBA.UtilClass=com.ibm.CORBA.iiop.Util \
 -Dcom.ibm.CORBA.iiop.noLocalCopies=true \
 -DDER_DRIVER_PATH=$DER_DRIVER_PATH \
 com.ibm.ws.bootstrap.WSLauncher \
 com.ibm.ejs.sm.util.process.Nanny $WAS_HOME/bin/admin.config}
```

Change this section as shown below. New parts are indicated with bold text.

```
if [ "${DB_TYPE}" != "DB2" ]
then
{
  ORACLE_HOME=$DB_instance_home
  export ORACLE_HOME
  LD_LIBRARY_PATH=$WAS_HOME/lib/odbc/lib:$WAS_HOME/bin:
 LD_LIBRARY_PATH=$WAS_HOME/bin:$WAS_HOME/lib:$DB_INSTANCE_HOME/lib:$LD_LIBRARY_PATH
 LIBPATH=$WAS_HOME/bin:$WAS_HOME/lib:$DB_INSTANCE_HOME /lib:$LIBPATH
  export LD_LIBRARY_PATH LIBPATH
  NLS_LANG=LANGUAGE_TERRITORY.UTF8
  export NLS_LANG

 ${JAVA_EXE?} \
 -classpath $WAS_HOME/lib/bootstrap.jar:$CLASSPATH \
 -Dws.ext.dirs=$WAS_EXT_DIRS \
 -Djavax.rmi.CORBA.UtilClass=com.ibm.CORBA.iiop.Util \
 -Dcom.ibm.CORBA.iiop.noLocalCopies=true \
 -DDER_DRIVER_PATH=$DER_DRIVER_PATH \
 com.ibm.ws.bootstrap.WSLauncher \
 com.ibm.ejs.sm.util.process.Nanny $WAS_HOME/bin/admin.config}}
```

where *DB_instance_home* is the fully qualified path to your Oracle home
directory; for example, /oracle/u01/app/oracle/product/8.1.7, and
LANGUAGE_TERRITORY is your language and territory as selected from the
following list:

- AMERICAN_AMERICA
- BRAZILIAN PORTUGUESE_BRAZIL
- FRENCH_FRANCE
- GERMAN_GERMANY
- ITALIAN_ITALY
- JAPANESE_JAPAN
- KOREAN_KOREA
- SIMPLIFIED CHINESE_CHINA
- SPANISH_SPAIN
- TRADITIONAL CHINESE_TAIWAN

6. Save your changes and exit the text editor.

Testing the installation: This section describes how to test the installation and
configuration of your WebSphere Application Server system. These instructions
assume that you have installed a supported Web server, database, and the
WebSphere Application Server component.

Perform the following steps to test your WebSphere installation:

1. Ensure that you are logged into the machine with superuser (root) privileges.
2. Start the WebSphere administrative server by executing the startupServer
   script, as follows:

```
cd /usr/WebSphere/AppServer/bin
./startupServer.sh
```

Ensure that the administrative server has started successfully by checking the file named `tracefile` located in the `/usr/WebSphere/AppServer/logs` directory.

The message `Server open for e-business` appears in this file when the server has started successfully.

3. Start the administrative console by running the `adminclient` script, as follows:

```
cd /usr/WebSphere/AppServer/bin
./adminclient.sh
```

4. When the console displays the message `Console Ready`, administer the application server by performing the following steps:

   a. When the administrative console opens, a tree view is displayed. Click the plus sign (+) next to **WebSphere Administrative Domain** entry to expand the view.

   b. Click the plus sign (+) next to the **Nodes** entry to expand the view.

   c. Identify the name of your host machine and expand the view of that entry.

   d. Click the plus sign (+) next to the **Application Servers** entry to expand the view.

   e. Select the **Default Server** entry and click the **Start** located on the tool bar. An information window opens and indicates that the server has started. Click **OK** to close the information window.

      After the default server is started initially, it will start automatically if it stops or if you restart the machine. If the administrative server fails, the default server continues to run.

5. Ensure that the Web server is running. If the Web server is not running, start it.

6. Start a browser and enter the URL for the snoop servlet, which is a sample servlet that is installed by default, as follows:

```
http://machine_name/servlet/snoop
```

where *machine_name* represents the name of the machine on which WebSphere Application Server is running. Information about `/servlet/snoop` is displayed.

If the IBM HTTP Server does not work normally, refer to the chapter "Enabling SSL for production with IBM HTTP Server" in the *WebSphere Commerce 5.4 Installation Guide* to enable SSL manually. You will not be able to successfully launch your store unless SSL is properly enabled.

Testing with an enterprise bean: This section describes how to test your WebSphere configuration by using an enterprise bean and the Increment sample. These instructions assume that you have installed your WebSphere Application Server system and have tested the installation as described in the previous section.

Perform the following steps to test your WebSphere configuration using an enterprise bean:

1. Ensure that you are logged into the machine with superuser (root) privileges.

2. Ensure that the administrative console is running.

3. Ensure that the default server (located under **WebSphere Administrative Domain > Nodes > node_name > Application Servers**) is running.

4. Ensure that the Web Server is running. If the Web server is not running, start it.

5. Start a Web browser and specify the following URL:

```
http://machine_name/webapp/examples/HitCount
```

where *machine_name* represents the name of the machine on which WebSphere is running. When the Web page opens, several selection options are displayed.

6. Under the heading **Generate hit count using**, click the radio button for the **Enterprise Java Bean** option.

7. Under the heading **Transaction Type**, click the radio button for the **None** option.

8. Click **Increment**.

If the number of hits is displayed, WebSphere is functioning properly.

To stop the Default Server, do the following:

1. Highlight the entry **Default Server** and click the **Stop** icon on the tool bar. An information window opens stating that the server has stopped.

2. Click **OK** to close this window.

Enable SSL: To enable Secure Sockets Layer (SSL) for IBM HTTP Server for testing purposes, use the `httpd.conf.sample` file as your IBM HTTP Server configuration file for testing purposes:

1. Switch to the `/usr/HTTPServer/conf` directory.

2. Back up the existing `httpd.conf` file by renaming it to `httpd.conf.bak`.

3. Rename `httpd.conf.sample` to `httpd.conf` .

4. If the following lines are missing from your `httpd.conf` file, add them to the end of the file or copy them over from the `httpd.conf.bak` file.

```
LoadModule ibm_app_server_http_module /usr/WebSphere/AppServer/bin/mod_ibm_app_server_http.so
WebSpherePluginConfig /usr/WebSphere/AppServer/config/plugin-cfg.xml
AddModule mod_app_server_http.c
```

To enable SSL for production purposes, refer to the chapter "Enabling SSL for production with IBM HTTP Server" in the *WebSphere Commerce 5.4 Installation Guide*.

Upgrading to WebSphere Commerce 5.4

This section describes how to upgrade to WebSphere Commerce 5.4 using the WebSphere Commerce 5.4 installation program.

Upgrading WebSphere Commerce Suite to WebSphere Commerce 5.4

This section explains how to install WebSphere Commerce Version 5.4. To complete the steps in this chapter, you will need the WebSphere Commerce Disk 1 CD and the WebSphere Commerce Disk 2 CD.

Important:

The following tasks must be completed before installing WebSphere Commerce Version 5.4:

- Ensure that you have installed:
  - A Web server
  - A database (either DB2 Universal Database or Oracle)
  - WebSphere Application Server

Installation procedures

> **Important**
>
> Do not delete your previous WebSphere Commerce Suite installation directory (/usr/lpp/CommerceSuite) as certain files need to be copied over from the previous installation directory to the new one.

To install WebSphere Commerce Version 5.4, do the following:

1. Log on as user ID `root`.
2. If necessary, mount the WebSphere Commerce Disk 1 CD. To do this, type

   `mount CDROM_dir`

   where *CDROM_dir* is the directory that you want to specify as your mounted CD. For instructions on allocating a CD file system, refer to the *WebSphere Commerce 5.4 Installation Guide*.
3. Ensure that your database has been started.
4. On an AIX command line, switch to the WebSphere Commerce install directory on the CD by typing:

   `cd /CDROM_dir/WebSphereCommerce`

   where *CDROM_dir* is the directory on which your CD is mounted.
5. On the command line, type `smitty install_all`.
6. In the **Input device/directory for software** field, type: **./** and press **Enter**.
7. Select **Software to install** and press F4=List to list the components available for installation.
8. To select WebSphere Commerce for installation, select the following from the list:
   - Commerce.Base:

     `5.4.0.0  Commerce Base Files`

     `5.4.0.0  Commerce license`

     `5.4.0.0  Commerce properties`

     `5.4.0.0  Commerce runtime`

     `5.4.0.0  Commerce samples`

     `5.4.0.0  Commerce schema`
   - Commerce.Blaze

     `5.4.0.0 Blaze Advisor Suite`
   - Commerce.Docs

     `5.4.0.0  Commerce xxxxxxxxx docs, readme and help`
     where *xxxxxxxxx* is a descriptive phrase for your language. For example, `5.4.0.0  WebSphereCommerce US english help and manuals`.
9. When you have made all your selections, press **Enter** to close the list.
10. In the **Detailed Output** field, press the **Tab** key to toggle from `No` to `Yes`. Press **Enter**.
11. To begin installation, press **Enter**.
12. A confirmation message appears. Press **Enter**.

The Command Status window appears, indicating that the installation of the selected components has started. The installation completes when the **Command** field at the top of the window changes from **Running** to **OK**.

Depending on the speed of your processor and the components you select, the install may take up to about 20 minutes.

13. When the installation has completed, scroll to the Installation section at the bottom of the listing. In the Result column, you should see either "Success" or "Already installed" next to the name of each component; if you do not, correct the problem and attempt the install again.

14. Press F10=Exit.

15. Unmount the CD by typing,

```
cd /
umount CDROM_dir
```

where *CDROM_dir* is the directory on which your CD is mounted.

16. Remove the WebSphere Commerce Disk 1 CD.

Upgrading your WebSphere Application Server version

You need to upgrade your software levels to meet the requirements of WebSphere Commerce. To do this, do the following:

1. Ensure that WebSphere Application Server is stopped and all processes associated with your Web server are stopped.

2. If you are using DB2, ensure that your DB2 services have been stopped.

3. If necessary, mount the WebSphere Commerce Disk 2 CD. To do this, type

```
mount CDROM_dir
```

where *CDROM_dir* is the directory that you want to specify as your mounted CD.

4. Go to `/usr/WebSphere/CommerceServer/bin`

5. Type: `./wc54efixunix.sh`

To verify that the script upgraded your software with no errors, check the following error logs:

- `WC54efix.log` in either the `/tmp` or `/usr/WebSphere/CommerceServer/logs` directory.
- `/usr/WebSphere/AppServer/eFix/PQ54291/Extractor.log`

> **Important**
>
> + After you install WebSphere Commerce 5.4, you need to install a migration
> + bundle that is available from the WebSphere Commerce Support site
> + (`http://www.ibm.com/software/commerce/support/`). Click on the link for the
> + edition you are using and see the section **Software downloads**. Download the
> + migration bundle, APAR IY46716 and follow the instructions that are
> + provided in the bundle to install it.
> +
> + You should not proceed with the migration steps in this Guide (such as
> + migrating your instance or database) without installing this migration bundle.

Upgrading to Payment Manager 3.1.2

Prior to upgrading to Payment Manager 3.1.2, consult the latest WebSphere Payment Manager product information available at:

http://www.ibm.com/software/webservers/commerce/payment

Before migrating to Payment Manager 3.1.2, back up your existing WebSphere Payment Manager database if you have not backed it up already. For information on how to back up your database, see "Backing up your databases" on page 28. After backing up your database:

- Read the latest README file, readme.framework.html, accessed through the documentation library link on the WebSphere Payment Manager Web site at:

    http://www.ibm.com/software/webservers/commerce/payment

- Remove any of the cassettes that you do not want to migrate before you install this new version of WebSphere Payment Manager.

The Payment Manager 3.1.2 installation and migration program will migrate data from Payment Manager, Version 2.1.4.0 or higher. The existing database will continue to be used, and any required data migration will occur during the installation.

Migration is possible only from a working version of WebSphere Payment Manager, Version 2.1.4.0 or higher. That is, migration from WebSphere Payment Manager, Version 2.1.3 or earlier modification levels is not supported. If you are using Payment Manager, Version 2.1.3 or earlier, you must upgrade to Version 2.1.4.0 before you can migrate to WebSphere Payment Manager, Version 3.1.

Payment cassettes

All cassettes (IBM provided or third party) previously installed on Payment Manager, Version 2.1 or higher are not guaranteed to function properly after installing Payment Manager 3.1.2. Before you install WebSphere Payment Manager, note the following:

- If you are currently using third-party payment cassettes, you should first check with your cassette provider to ensure your existing cassette can be migrated to Payment Manager 3.1.2 before attempting to install the WebSphere Payment Manager product on your system.

- Both third-party and IBM-provided payment cassettes must be migrated to Payment Manager 3.1.2 for the cassettes to function properly. If you are using IBM-provided cassettes (such as the Cassette for VisaNet or Cassette for SET™), you must install the version 3.1.2 cassette software to perform migration. For installation instructions, refer to the cassette-specific supplement for your payment cassette.

- If you are currently using WebSphere Payment Manager Version 2.1.4 with the Cassette for SET Version 2.1.4, you must first install the Version 2.1.5 PTFs (for both the WebSphere Payment Manager framework and the Cassette for SET). Then, start the Payment Engine at least once, and stop the engine, before migrating. Otherwise, the installation of Payment Manager 3.1.2 will fail during the data migration.

Databases

If you are migrating from a prior installation, ensure your database product, operating system, and companion products are at a level supported by Payment

Manager. For platform-specific information, see the installation chapter corresponding to your platform in the *IBM WebSphere Payment Manager 3.1 for Multiplatforms Installation Guide*.

If you are using a version of UDB earlier than DB2 7.2 FixPak 5, you must upgrade to this version of UDB. For information on upgrading your UDB software from CD-ROM, see the section "Installing a database (UDB) on workstation platforms" in the *IBM WebSphere Payment Manager 3.1 for Multiplatforms Installation Guide*. Optionally, if your Payment Manager installation used a database other than UDB and you now want to change to UDB, you may manually install UDB and migrate your WebSphere Payment Manager data to UDB.

Premigration considerations

Note the following additional considerations before migrating:

Paths in registry keys
If you have upgraded or reinstalled any of the WebSphere Payment Manager prerequisites (Web server, WebSphere Application Server, database), ensure that the paths saved in the WebSphere Payment Manager or Payment Server registry keys are correct for your current installation. These keys can be found in the file `/usr/bin/jitk.db`.

If the paths in the following keys do not reflect where the WebSphere Payment Manager prerequisites are installed on your system, update them with the paths appropriate for your system. The paths shown in the following examples reflect registry keys on Windows. Use the appropriate syntax for your platform when updating the values.

Check for the following:
- If you have upgraded or reinstalled your database, check:
  ```
  21db_jdbcClasspathAddition="/home/db2_instance_name/sqllib/bin/java12/db2java.zip"
  21db_jdbcSharedLibPathAddition="/home/db2_instance_name/sqllib/bin"
  21db_odbcSharedLibPathAddition="/home/db2_instance_name/sqllib/bin"
  ```
- If you have IBM cassettes installed, also check the registry keys that have the same name as the above with a prefix specific to each cassette. For example, if you have the Cassette for CyberCash installed, check:
  ```
  CC_21db_jdbcClasspathAddition="/home/db2_instance_name/sqllib/bin/java12/db2java.zip"
  CC_21db_jdbcSharedLibPathAddition="/home/db2_instance_name/sqllib/bin"
  CC_21db_odbcSharedLibPathAddition="/home/db2_instance_name/sqllib/bin"
  ```

  When you update the `21db_jdbcClasspathAddition` key and any related cassette keys, make sure to use the JDBC 2.0 version of the database driver. For example, change this:
  ```
  21db_jdbcClasspathAddition="/home/db2_instance_name/sqllib/bin/java/db2java.zip"
  ```

  to the following:
  ```
  21db_jdbcClasspathAddition="/home/db2_instance_name/sqllib/bin/java12/db2java.zip"
  ```
- If your database is UDB, also check:
  ```
  UDB_DIR="/home/db2_instance_name/sqllib"
  ```
- If you have upgraded or reinstalled your Web server, check:
  ```
  WEB_PUBDIR="/usr/HTTPServer/htdocs"
  ```
- If you have upgraded or reinstalled WebSphere Application Server, check:

```
WAS_dir="/usr/WebSphere/AppServer"
WAS_ver="4.0.1"
javaExe="/usr/WebSphere/AppServer/java/jre/bin/java"
javaClasses="/usr/WebSphere/AppServer/java/jre/lib/rt.jar;
  /usr/WebSphere/AppServer/java/jre/lib/i18n.jar"
```

- You should also update the `ibmpayserver.sh` script to ensure that references to the WebSphere Application Server directory are correct, for example, `/usr/WebSphere/AppServer/java/jre/bin/java` directory.

Test Cassette

The Test Cassette provided in WebSphere Payment Manager Version 2.1.x is no longer provided, starting with Version 2.2. The installation program will delete the Test Cassette if found. If you are currently using the Test Cassette, it is recommended that you convert to using the OfflineCard Cassette.

For more information about the OfflineCard Cassette, see the *WebSphere Payment Manager Administrator's Guide*.

EventListener SocksHost length limitation

The length of the SocksHost field is now limited to 254 characters. If an EventListener has been created by an application with a SocksHost field with a length greater than 254 characters, it will be truncated to 254 characters by the migration program. For more information on event notification, see the *WebSphere Payment Manager Programmer's Guide*.

PSWAS2Realm Migration

If your WebSphere Payment Manager environment used the `PSWAS2Realm` (as identified in the `PaymentServlet.properties` file), the realm is not migrated when you install Payment Manager 3.1.2. WebSphere Payment Manager will use `PSDefaultRealm` instead. No action on your part is necessary as a result of this change.

Installation and migration

The Payment Manager 3.1.2 installation and migration program will migrate data from WebSphere Payment Manager, Version 2.1.4.0 or higher. The existing database will continue to be used, and any required data migration will occur during the installation.

See the appropriate installation section of the *IBM WebSphere Payment Manager 3.1 for Multiplatforms Installation Guide* to install Payment Manager 3.1.2 for your platform.

Note: The *IBM WebSphere Payment Manager 3.1 for Multiplatforms Installation Guide* directs you to switch the `wpm.RealmClass` parameter from `com.ibm.etill.framework.payserverapi.PSDefaultRealm` to `com.ibm.commerce.payment.realm.WCSRealm`. This tells Payment Manager to use WebSphere Commerce to authenticate the user during log on. That means WebSphere Commerce must be running and the user ID used logging in to WebSphere Payment Manager must be a valid WebSphere Commerce administrator ID.

It is recommended that you log into the WebSphere Commerce Administration Console at least once before using WebSphere Payment Manager. To log into the WebSphere Commerce Administration Console, go to:

`https://host_name:8000/adminconsole`

Enter the default Administration Console user ID (`wcsadmin`), and the default password (`wcsadmin`). You will be asked to change your password the first time you log in.

Postmigration considerations

Cassettes

After a successful installation and migration of the WebSphere Payment Manager Framework, you must upgrade your IBM cassettes and Version 2.x non-IBM cassettes to version 3.1.2 for the cassettes to function properly.

Removed files

All of the PSPL files that contain a language qualifier (represented by **) in the name are deleted for the WebSphere Payment Manager and IBM WebSphere Payment Manager cassettes including:

```
pspl/admin.**.PSPL
pspl/payment.**.PSPL
pspl/reports.**.PSPL
```

Language-qualified PSPL files for third-party cassettes are not removed.

Because Tivoli® Ready support for WebSphere Payment Manager is no longer provided, the *PMInstallDir*/tivsupport directory is removed.

In addition:

- After a successful install, the entire `jre` subdirectory is removed (WebSphere Payment Manager uses the IBM SDK for Java that WebSphere Application Server uses).
- Web pages that were formerly listed under the Web publish directory *WebPubDir* are removed and made available in the `WebSphere Payment Manager.war` file contained in the `PaymentManager.ear` file located in the *WAS_Home*/installedApps directory.
- The `FormatServletTrace.cmd` file, which was previously used to format trace log information, is removed. You can format trace information with the `FormatTrace.cmd` file instead. For more information about this command file, refer to the *WebSphere Payment Manager Administrator's Guide*.

Backed up files

During migration, these key files and subdirectories are saved to a new backup directory: *<pmInstallDir>*/!pm22Backup! or *<pmInstallDir>*/!pm21Backup! (depending on what version you had previously):

```
.payment
IBMPayServer [.cmd ]
IBMPaymentServerUI.properties
PaymentServlet.properties
PSDefaultRealm
StopIBMPayServer [.cmd ]
log/*(all files within this directory)
samples/*(all files within this directory)
```

Upgrading UDB

If you are using UDB as your database manager product, you must ensure that you are using the level of UDB required for this version of WebSphere Payment Manager (DB2 7.2 FixPak 5 or higher). If you need to upgrade to this level of UDB, see the section "Upgrading UDB" in the *IBM WebSphere Payment Manager 3.1 for Multiplatforms Installation Guide*.

If you are installing UDB for the first time and no migration is necessary, see
"Installing a database (UDB) on workstation platforms" in the *IBM WebSphere
Payment Manager 3.1 for Multiplatforms Installation Guide* for installation procedures.

Running the post installation script

After you have installed WebSphere Commerce and all its required components,
you must run the post installation script. To run the script complete the following
steps:

1. Ensure that your DB2 services have been stopped.
2. Ensure that WebSphere Application Server is stopped and that all processes
   associated with your Web server are stopped.
3. Open a command window.
4. Change to the /usr/WebSphere/CommerceServer/bin directory.
5. Run the script as follows (without any parameters):

   `./wcpostinstall.sh`
6. Follow the on screen prompts. It is recommended that you type `yes` when
   asked if you want to run as a non-root user.

   Notes:

   a. If you choose to run WebSphere Application Server under the root user,
      then the root user will own both the WebSphere Application Server and
      WebSphere Commerce install trees. If you choose to run WebSphere
      Application Server under a non-root user, then a non-root user (by default,
      `wasuser`) will own the WebSphere Application Server, WebSphere Payment
      Manager, and WebSphere Commerce installation trees.

      If you choose to run WebSphere Application Server under the root user, you
      should run all steps in the migration process as the root user. If you choose
      to run WebSphere Application Server under the non-root user (`wasuser`),
      you should run all steps in the migration process as the non-root user.

   b. It is also recommended that you accept the default group and user names
      used by the `wcpostinstall.sh` script. If you choose different names you
      must remember to substitute those names throughout the documentation.

   c. Capital letters are not supported in group or user names.

The script does the following:

- Sets up the `product.xml` file for WebSphere Commerce 5.4 with the required
  fields (if your previous WebSphere Commerce Suite 5.1 installation directory is
  available on the same machine).
- Updates the `setenv.sh` script to have the `WAS`, and the `DB2` or `Oracle` Home
  values required for proper operation.
- Modifies the `default.xml` as well as the `populatedb` and `populatedbnl` scripts for
  the `LOCALE` settings
- Sets up the access permissions for files and directories in the WebSphere
  Commerce 5.4 tree in order to limit access to files.

Updating product.xml

If your previous version of WebSphere Commerce is not on the same machine as WebSphere Commerce 5.4 when migrating, you will need to edit the `product.xml` file on the WebSphere Commerce 5.4 machine and manually set the values. This step is required if you are following the procedures in "Minimizing downtime during migration" on page 15. If your previous version of WebSphere Commerce is on the same machine, you can skip this step.

Modify the `<migrationFrom>` section in the `product.xml` file in the `/usr/WebSphere/CommerceServer/xml` directory as follows. (The `<migrationFrom>` section specifies information for your previous machine, from which you are migrating.)

```
<migrationFrom>
   <edition>
      <name>name</name>
   </edition>
   <version>5</version>
   <release>rel</release>
   <modification>mod</modification>
   <fixpak>0</fixpak>
   <path>path</path>
</migrationFrom>
```

In the above:

- *name* is one of:

  Start   WebSphere Commerce Suite Start Edition — Windows or Linux (Intel) only

  Pro   WebSphere Commerce Suite Pro Edition

  Business   WebSphere Commerce Business Edition 5.1 — Windows only

  Business Limited Availability
        (WebSphere Commerce 5.4 Limited Availability release)

- *rel* is one of:

  1   For:
  - WebSphere Commerce Suite 5.1.1 — Windows only
  - WebSphere Commerce Suite 5.1
  - WebSphere Commerce Business Edition 5.1 — Windows only
  - WebSphere Commerce Suite 5.1.1.1
  - WebSphere Commerce Suite 5.1.1.2

  4   For WebSphere Commerce 5.4 Limited Availability release

- *mod* is one of:

  0.1   WebSphere Commerce Suite 5.1.0.1

  1   WebSphere Commerce Suite 5.1.1 — Windows only

  1.1   WebSphere Commerce Suite 5.1.1.1

  1.2   WebSphere Commerce Suite 5.1.1.2

- *path* is the installation path for the previous Commerce system, for example, /usr/lpp/CommerceSuite for WebSphere Commerce Suite 5.1.

The following is an example of what your `product.xml` file needs to contain if you are migrating from WebSphere Commerce Suite 5.1.0.1 Pro Edition:

```
<migrationFrom>
   <edition>
      <name>Pro</name>
   </edition>
   <version>5</version>
   <release>1</release>
   <modification>0.1</modification>
   <fixpak>0</fixpak>
   <path>/usr/lpp/CommerceSuite</path>
</migrationFrom>
```

Updating migrate.xml (when migrating from WebSphere Commerce 5.1.1.2)

If you have applied WebSphere Commerce FixPak 2 to your WebSphere Commerce 5.1 system, and are thus at the WebSphere Commerce 5.1.1.2 level, you need to update the `<MigratePath>` section in the `migrate.xml` file (located in the `/usr/WebSphere/CommerceServer/xml/config/migrate` directory) as indicated in bold below, in order to specify the correct modification level:

```
<MigratePath OS="Win,AIX,Sun,Linux">
   <From version="5"
         release="1"
         modification="1.2"
         fixpak="0" />
   <To version="5"
       release="4"
       modification="0"
       fixpak="1"
       edition="Business" />
   <MigrateAction>
    <WASMigrate migrate="true" />
      <WSMigrate migrate="true" />
      <PMMigrate migrate="true" />
      <WCSMigrate migrate="true" >
         <migrateFile file="$INSTALL_DIR$/xml/config/migrate/common/WCSmigrate511_51.xml" />
         <migrateFile file="$INSTALL_DIR$/xml/config/migrate/common/WCSmigrate51BE_54LA.xml" />
         <migrateFile file="$INSTALL_DIR$/xml/config/migrate/common/WCSmigrate54LA_54.xml" />
      </WCSMigrate>
   </MigrateAction>
 </MigratePath>
```

Chapter 4. Migrating your database

This chapter describes the steps to migrate your Commerce Suite 5.1 database schema to the WebSphere Commerce 5.4 schema level. Prior to performing the steps to migrate your schema, it is recommended that you have completed the steps in Chapter 2, "Backing up Commerce Suite 5.1," on page 27 and Chapter 3, "Upgrading your software," on page 35.

> **Important**
>
> Before proceeding with the migration steps described in this chapter and the remainder of this Guide:
>
> - If you have not already done so, download the latest WebSphere Commerce 5.4 FixPak from the WebSphere Commerce Downloads Web site.
>   - For Business Edition:
>
>   `http://www.ibm.com/software/webservers/commerce/wc_be/downloads.html`
>   - For Professional Edition:
>
>   `http://www.ibm.com/software/webservers/commerce/wc_pe/downloads.html`
>
>   You will need to apply this WebSphere Commerce 5.4 FixPak *after* you upgrade your software from WebSphere Commerce Suite 5.1 to WebSphere Commerce 5.4, but *prior* to migrating your assets, such as the commerce database, instance, configuration, and code. The FixPak contains corrections that will minimize any problems you may encounter in the migration process.
>
>   Refer to the installation instructions that accompany the FixPak for details on how to apply it.
>
>   The FixPak contains two scripts that you must run for migration, as documented in "Running the install_wc script" and "Running the install_was script" on page 103.
> - To run the following database migration utilities, the DB2 database has to be cataloged properly if it is a remote database. If it is local, the database will be cataloged by default.
>
>   If you are using a remote Oracle database, you need to establish connectivity to the remote Oracle server as described below. If you are using a local Oracle database, you can ignore this requirement.
> - If you have extended the WebSphere Commerce Suite schema (for example, by adding a column to a table) you should perform the steps documented in Appendix B, "Extending the data migration script," on page 157.
> - WebSphere Commerce summary tables are not loaded by the WebSphere Commerce 5.4 installation or migration process. If you wish to load the summary tables after migration, you can load them using the Search Configuration panel in the WebSphere Commerce Configuration Manager after you have completed migration. See the Configuration section of the *WebSphere Commerce 5.4 Installation Guide*.
> - Ensure that you have enough free disk space on your machine to accommodate backups of the Commerce Suite 5.1 database. This additional disk space should be at least three times the size of your current WebSphere Commerce Suite 5.1 database, as the database migration script will perform three backups of the database during migration.

Running the install_wc script

Before performing the database migration task, you need to run the `install_wc.sh` script that is provided with the WebSphere Commerce 5.4 FixPak in order to minimize any problems with the migration.

Before running the script, shut down any Web servers that are running as well as WebSphere Application Server.

To run the script, go to the directory where you have unpackaged the files provided with the WebSphere Commerce 5.4 FixPak and enter:

```
./install_wc.sh WC_install_path WAS_install_path
```

where:
- *WC_install_path* is the WebSphere Commerce installation path (for example, /usr/WebSphere/CommerceServer).
- *WAS_install_path* is the WebSphere Application Server installation path (for example, /usr/WebSphere/AppServer).

Setting up the STORE table and populatedb script

If you are using a national language version of WebSphere Commerce, ensure that `LANGUAGE_ID` for the default store is set correctly to your national language in the `STORE` table as well as in the `populatedb` script, prior to data migration.

1. In the `STORE` table, check that the `LANGUAGE_ID` is set to the correct value for your national language for the default store (that is, the store where `STORE_ID=0`):

LANGUAGE_ID value	Language
-1	English (US)
-2	French
-3	German
-4	Italian
-5	Spanish
-6	Brazilian Portuguese
-7	Simplified Chinese
-8	Traditional Chinese
-9	Korean
-10	Japanese

2. Backup the original `populatedb` script that is located in the /usr/WebSphere/CommerceServer/bin directory:
   - For DB2 databases — `populatedb.db2.sh`
   - For Oracle databases — `populatedb.oracle.sh`

3. Copy the `wcs.comment_multi_en_US.xml` to a locale-specific file name in the/usr/WebSphere/CommerceServer/schema/xml directory. For example, if you are using the Japanese locale, ja_JP, copy the file `wcs.comment_multi_en_US.xml` to `wcs.comment_multi_ja_JP.xml`.

4. Modify the `populatedb` script that you backed up to ensure that all locale information is correct for your national language (for example, `ja_JP` for Japanese). Search for *all* occurrences of the English (United States) locale — `en_US` in the script.

   ```
   .
   .
   .
   %JAVA_HOME%bin/java -Dcom.ibm.wca.logging.configFile=%wcaloggerconfigfile%
     -Dcom.ibm.wcm.ErrorReporterDir=%errorlogdir% -classpath %CP% %massLoader%
     -infile %schema%/xml/wcs.bootstrap_en_US.xml -dbname %database%
     -dbuser %user% -dbpwd %password% -method sqlimport -commitcount 100 -schemaname
     %schemaowner% >> %log%
   %JAVA_HOME%/bin/java -Dcom.ibm.wca.logging.configFile=%wcaloggerconfigfile%
     -Dcom.ibm.wcm.ErrorReporterDir=%errorlogdir% -classpath %CP% %massLoader%
     -infile %schema%/xml/wcs.comment_multi_en_US.xml -dbname %database%
     -dbuser %user% -dbpwd %password% -method sqlimport -commitcount 100 -schemaname
   ```

```
%schemaowner% >> %log%
.
.
.
```

Note that in the above example, only two occurrences are shown; in actuality there will be many occurrences.

Change *all* occurrences of en_US to the locale for your national language. For example, if your national language is Japanese, and LANGUAGE _ID=-10 where STORE_ID=0 in the STORE table, change all occurrences of en_US to ja_JP as indicated in bold for the populatedb script as follows:

```
%JAVA_HOME%bin/java -Dcom.ibm.wca.logging.configFile=%wcaloggerconfigfile%
  -Dcom.ibm.wcm.ErrorReporterDir=%errorlogdir% -classpath %CP% %massLoader%
  -infile %schema%/xml/wcs.bootstrap_ja_JP.xml -dbname %database%
  -dbuser %user% -dbpwd %password% -method sqlimport -commitcount 100 -schemaname
  %schemaowner% >> %log%
%JAVA_HOME%/bin/java -Dcom.ibm.wca.logging.configFile=%wcaloggerconfigfile%
  -Dcom.ibm.wcm.ErrorReporterDir=%errorlogdir% -classpath %CP% %massLoader%
  -infile %schema%/xml/wcs.comment_multi_ja_JP.xml -dbname %database%
  -dbuser %user% -dbpwd %password% -method sqlimport -commitcount 100 -schemaname
  %schemaowner% >> %log%
```

Note, that there may be more occurrences than shown in the above example.

The following Java locales are supported by WebSphere Commerce 5.4.

Locale name	Political, geographical, or cultural region
en_US	United States
fr_FR	France
de_DE	Germany
it_IT	Italy
es_ES	Spain
pt_BR	Brazil
zh_CN	China
zh_TW	Taiwan
ko_KR	Korea
ja_JP	Japan

5. Ensure that in the STORE table, the LANGUAGE_ID for your default store is the correct locale for your national language you want:

```
DB2 CONNECT TO database
DB2 SELECT LANGUAGE_ID FROM STORE WHERE STORE_ID=0
```

For example, if your national language is Japanese, and the LANGUAGE_ID is not correct for Japanese, say it is set to -1 (which is en_US), you would need to modify it manually to −10 (for ja_JP)

```
DB2 UPDATE STORE SET LANGUAGE_ID=-10  WHERE STORE_ID=0
```

6. Perform the database migration as described in the remainder of this chapter. Once you have completed the database migration, set the LANGUAGE_ID back to whatever it was originally, in our example, -1 for en_US:

```
DB2 UPDATE STORE SET LANGUAGE_ID=-1 WHERE STORE_ID=0
```

Cataloging a remote DB2 database (DB2–only)

If you are using a remote DB2 database, catalog it on your local node as follows:

1. Switch to the DB2 instance user ID for your database (for example, db2inst1):

```
su - db2_instance_user_id
```

2. Catalog the remote database server node by typing the following:

```
db2 catalog tcpip node node_name
 remote database_server_hostname server port_number
```

where:

node_name
> Is the unique name of your choice that DB2 will use to identify the TCP/IP node.

database_server_hostname
> Is the TCP/IP host name of your database server.

port_number
> Is the port being used by DB2. (At installation time, DB2 attempts to use 50000 by default. If it cannot use this port number, it will pick an unused port that is available.)

3. Catalog the remote DB2 database on the remote node:

```
db2 catalog db db_name at node node_name
```

where:

db_name
> Is your DB2 database (for example, `mall`).

node_name
> Is the remote node you cataloged above.

Scenario to avoid

During migration, avoid the following type of database usage scenario:

Suppose on one machine called hostA, you have two database instances, one is called db2inst1 which is the default instance, and the other is called ecom. Under instance db2inst1, there is a database named dbtest.

On another machine, say hostB, you also have two database instances, one is db2inst1 and the other is called ecom. Under instance ecom, there is a database named dbtest (the database user and password is the same as the database dbtest on hostA).

On machine hostA, do the following:

```
su - ecom
db2 catalog tcpip node hostB remote hostB server 50000
db2 catalog db dbtest as dbtest at node hostB
```

Thus on hostA, you will have a remote database named dbtest which is redirected to hostB.

Under this condition, if you running WebSphere Commerce on hostA, the database to which you are actually connected is not dbtest on hostB, but dbtest on hostA. Because the default instance is db2inst1, WebSphere Commerce will use this instance name to connect to database.

Establishing Oracle8i client and server connectivity (Oracle-only)

If you are using a remote Oracle database, you need to establish connectivity to the Oracle8i Server where your data resides. It is assumed that you have installed the Oracle 8.1.6 Client in accordance with the Oracle 8.1.6 product installation information on the WebSphere Commerce 5.4 machine.

Note that you need to update the JDBC™ driver in your Oracle installation
(`classes12.zip` if you are using JDK™ 1.2) to the latest available driver from the
Oracle Technet site:

`http://technet.oracle.com`

To establish connectivity with the Oracle8i server where your Commerce Suite data
resides, you need to add the following entries into the `tnsnames.ora` file that is
usually located in the */oracle*/ora81/network/admin directory.

In the following example, you need to update the italicized items for your own
environment:

```
<data_source_name> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <server_hostname>)(PORT = <port_number>))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = <Oracle_Service_ID>)
    )
  )
```

where:
- *<data_source_name>* is your data source on the Oracle client machine (for
  example, `wcs51`)
- *<server_hostname>* is your fully qualified Oracle server host name where your
  Commerce Suite 5.1 data resides (for example, `cs51host.toronto.ibm.com`)
- *<port_number>* is the port number for the Oracle server machine (for example,
  1521 is the default)
- *<Oracle_Service_ID>* is the Oracle SID for Commerce Suite 5.1 (for example, `mall`)

Please refer to Oracle documentation for further information on establishing
connectivity to the server.

Running the database preparation script

This script will modify the access control tables in your database so that they can
be properly migrated. Thus, you must run this script for proper database
migration.

The database preparation script will also analyze your existing Commerce Suite
databases and generate a report about any data with the following characteristics:
- All items that do not have a parent product
- All members that do not have a parent in the organization entity.

The report lists the items described in "Mandatory database premigration items"
on page 76 and "Optional database premigration items" on page 77.

For the mandatory items, you need to take the requested actions before proceeding
with the database migration.

For the optional items, you are not required to complete the recommended actions
before proceeding with the database migration. It is recommended, however, that
you correct any of the data that gets flagged or if you do not want the default
assignments of the migration script. After correcting your data, you should rerun
the database preparation script to ensure all items are fixed.

Once you have fixed all necessary items, back up your Commerce Suite databases once again as described in Chapter 2, "Backing up Commerce Suite 5.1," on page 27, in order that you have an up-to-date copy of your databases.

Run the WebSphere Commerce 5.4 database preparation script as described below.

DB2 databases

__ 1. Open a command prompt window.

__ 2. If you are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, `db2inst1`):

```
su - db2_instance_user_id
```

If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

```
su - WAS_user
```

__ 3. Switch to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

__ 4. Run the database preparation script as follows:

```
./premigrate.db2.sh db_name db_userID db_userID_password db2insthome
```

where:

- *db_name* is the Commerce Suite database that you are migrating (for example, `mall`).
- *db_userID* is the user ID to connect to the Commerce Suite database that you are migrating (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the Commerce Suite database that you are migrating (for example, `mypasswd`).
- *db2insthome* is the home directory for your DB2 instance user ID (for example, /home/db2inst1/sqllib)

For example, to run the database preparation script, you can use the following command:

```
./premigrate.db2.sh mall mydbuser mypasswd /home/db2inst1/sqllib
```

__ 5. Check the `premigrate.log` file generated in the `/usr/WebSphere/CommerceServer/logs/migration` directory for additional messages or errors. You need to resolve any errors that are flagged before proceeding.

If you have fixed the errors or warnings that are flagged by the database preparation script or made any changes to your database, it is recommended that you take another back up of your database as described in "Backing up your databases" on page 28.

Oracle databases

__ 1. Open a command prompt window.

__ 2. If you are running WebSphere Application Server as the root user, ensure you are logged in as the root user.

If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

```
su - WAS_user
```

__ 3. Switch to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

__ 4. Run the database preparation script as follows.

```
./premigrate.oracle.sh db_name db_userID db_userID_password
  host_name port_number
```

where:

- *db_name* is the Commerce Suite database that you are migrating (for example, `mall`).
- *db_userID* is the user ID to connect to the Commerce Suite database that you are migrating (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the Commerce Suite database that you are migrating (for example, `mypasswd`).
- *host_name* is the fully qualified host name of your machine.
- *port_number* is the Oracle listener port number. Typically, this is port 1521 by default.

For example, to run the database preparation script, you can use the following command:

```
./premigrate.oracle.sh mall mydbuser mypasswd myhost.montreal.ca 1521
```

__ 5. Check the `premigrate.log` file generated in the `/usr/WebSphere/CommerceServer/logs/migration` directory for additional messages or errors. You need to resolve any errors that are flagged before proceeding.

If you fix the errors or warnings that are flagged by the database preparation script or make any changes to your database, it is recommended that you take back up of your database as described in "Backing up your databases" on page 28.

Mandatory database premigration items

If the database preparation script generates the following return codes, you must take the following actions before proceeding with database migration.

Return Code   **Explanation/Action**

122   The script checks if your Commerce Suite 5.1 database uses the following member IDs (0 to -8) for the corresponding member group (for example, whether the member ID is **-1** for the Site Administrator member group).

0	Site Owner
-1	Site Administrator
-2	Customer
-3	Customer Service Representative
-4	Merchant
-5	Order Clerk
-6	Store Administrator
-7	Store Developer
-8	Merchandising Manager

Notes:

1. You are required to preserve all bootstrap values or else the database migration will fail.

2. Return code 122 is only applicable for English systems. If you are using a national language version other than English, you can ignore this return code for your system. National language version users need to make sure that they have not modified the bootstrap data.

3. In WebSphere Commerce 5.4, the Merchant role has been renamed to Seller, and Merchandising Manager renamed to Product Manager.

200   The script checks REFCODE field for the AUCTION table in Commerce Suite 5.1. There is a REFCODE field which is supposed to be unique, but the Commerce Suite 5.1 schema does not enforce it. The WebSphere Commerce 5.4 schema specifies that it is a unique index.

You are required to make sure that the REFCODE field in the AUCTION table is unique.

318   The script checks that your contract name is not more than 200 characters in length.

You are required to make sure that the data in the NAME field in the AUCTION table does not exceed 200 characters.

319   The script checks that the ORGENTITY table description is no more than 512 characters in length.

You are required to make sure that the data in the DESCRIPTION field in the ORGGRP table does not exceed 200 characters.

340   The script checks that the MBRGRP table description is no more than 512 characters in length.

You are required to make sure that the data in the DESCRIPTION field in the MBRGRP table does not exceed 200 characters.

Optional database premigration items

If the database preparation script generates the following return codes, it is recommended that you take the following actions before proceeding with database migration. Although these actions are not mandatory, you should carefully consider the implications on your migrated system.

Return Code   Explanation/Action

103   The script checks your Commerce Suite 5.1 discount data. If you have custom discount data that was created manually, that is, discount data that was not created using the **Merchandise** menu of the Commerce Suite Accelerator, the script will issue a warning. The database migration script will not migrate the discount data to WebSphere Commerce 5.4 requirements, however this data will remain in the database as-is. The WebSphere Commerce 5.4 discount tool will not be able to view the data.

If you wish to view this discount data (which is assumed to be created outside of the Commerce Suite 5.1 tools) you need to follow the same procedure you used before to handle your discount data in Commerce Suite 5.1.

+	305	The specified users have REGISTERTYPE of "A" (Administrative
+		user), but are not assigned to any access groups in the ACCMBRGRP
+		table. By belonging to access groups, the user may have privileged
+		access control rights in the site. Most administrative users belong
+		to one or more access groups in the ACCMBRGRP table. If you feel
+		that these users should belong to one or more access groups in the
+		ACCMBRGRP table, consider assigning the users to access groups in
+		the ACCMBRGRP table before migration.

307 If a user has a REGISTERTYPE value of S and it does not appear in the ACCMBRGRP table to have a more specific role for that user, the database migration script assigns a Site Administrator role to the user automatically during migration. Specifically, the migration script creates an entry for these users in the MBRROLE table, and assigns the Site Administrator role to their ancestor organization entries. Since Site Administrator is a very powerful role, the script issues a warning to advise you of this. The Site Administrator role should not be assigned to any organizational entities or users without careful consideration.

309 The Customer access group in Commerce Suite 5.1 represented the group of all users. An AllUsers member group is shipped in WebSphere Commerce 5.4 which replaces the Customer access group. If a user was assigned to the Customer access group (-2) in Commerce Suite 5.1, the migration script will explicitly assign that user to the AllUsers member group in WebSphere Commerce 5.4. The premigration script issues a warning message as such explicit assignment is not required in the WebSphere Commerce 5.4 design.

Assess whether you require such explicit assignment.

310 Commerce Suite 5.1 supported an Order Clerk role which is no longer required or used in WebSphere Commerce 5.4. The tasks that used to be performed by the Order Clerk role have either been automated or can be performed by the Customer Service Supervisor in WebSphere Commerce 5.4. If a user has an Order Clerk role (-5) in Commerce Suite 5.1 and an entry in the ACCCMDGRP table, the user is migrated as part of the access control migration and the role is treated as a user-defined role. If no users have the Order Clerk role, the role will not be migrated.

Review if you still require the Order Role clerk. If you do not require it, remove it since it is no longer supported in WebSphere Commerce 5.4.

316 The script checks for presence of parent member identifier for organizational entities in the ORGENTITY table.

+		You may want to fill in the parent MEMBER_ID in the ORGENTITY table
+		for the items that are flagged. Otherwise, the database migration
+		script will assign a value of -2001, which is the Root Organization.

401 The script checks for all the items that do not have a parent product. WebSphere Commerce 5.4 requires that each item have one parent product.

For any items that do not have a parent, you should add the item to the CATGPENREL table and assign a CATALOG_ID and CATGROUP_ID.

You can create a parent product for the items that are flagged. If you do not create a parent product for the items which are flagged, the database migration script will create them for you.

404   The script checks for all the items have more than one parent product. WebSphere Commerce 5.4 requires that each item can only have one parent product.

If you want to use the WebSphere Commerce 5.4 Catalog tools to view catalog data, you need to remove all parent products except for one parent product.

414   There are problems related to access control. Locate and refer to the detailed message in the log where Access Control checking messages are located. The following are examples of such messages:

- The user *xxxx* does not have an entry in the BUSPROF table. The user will be migrated to MBRROLE but will not actually be able to perform that role without being assigned to an organization or organization unit that can perform the role.

- The owner *xxxx* for the member group defined in ACCMBRGRP is not an organization and will not be migrated.

415   The script checks that your orders have a status code of M (Payment initiated - the customer has initiated payment. Authorization is in process).

Make sure that STATUS column in the ORDERS table for all orders is set to M.

416   The script checks that your order items have a status code of M (Payment initiated - the customer has initiated payment. Authorization is in process).

Make sure that STATUS column in the ORDERITEMS table for all order items is set to M.

Dropping Commerce Suite 5.1 cache triggers

If you have installed any cache triggers in Commerce Suite 5.1, you should drop them before migrating your database. You can use the wcs.cache.delete.trigger1.sql script provided in the following directory:

- For DB2 databases: /usr/WebSphere/CommerceServer/schema/db2
- For Oracle databases: /usr/WebSphere/CommerceServer/schema/oracle

DB2 Databases

To run the wcs.cache.delete.trigger1.sql script for DB2 databases:

1. Open a command prompt window.
2. If you are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, db2inst1):

   su - *db2_instance_user_id*

   If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

   su - *WAS_user*
3. Switch to the /usr/WebSphere/CommerceServer/schema/db2 directory.

4. Connect to your Commerce Suite database.

   ```
   db2 connect to db_name user db_user_name using db_password
   ```
5. Run the wcs.cache.delete.trigger1.sql script as follows:

   ```
   db2 -tvf wcs.cache.delete.trigger1.sql
   ```

Oracle databases

To run the wcs.cache.delete.trigger1.sql script for Oracle databases:

1. Open a command window.
2. Switch to the Oracle user ID (for example, oracle):

   ```
   su - oracle_user_ID
   ```
3. Switch to the /usr/WebSphere/CommerceServer/schema/oracle directory.
4. Enter the following command:

   ```
   sqlplus -s userid/password@service_name < wcs.cache.delete.trigger1.sql
   ```

   where:
   - *userid* is your Oracle user ID
   - *password* is the password for the Oracle user ID
   - *service_name* is the Oracle service name.

   For example:

   ```
   sqlplus -s oracle/oracle@o809 < wcs.cache.delete.trigger1.sql
   ```

Dropping custom constraints

If you have customized tables that contain foreign key links to the shipped
Commerce Suite 5.1 tables, the data migration script may fail when it attempts to
drop these referential integrity constraints (such as foreign keys, primary keys, and
indexes) during data migration. You need to drop these constraints using the SQL
statements in the sections below. You will later restore the constraints after
migrating your data to the WebSphere Commerce 5.4 schema as described in
"Restoring custom constraints" on page 85.

1. Identify all new tables that you have added as well as all the Commerce Suite
   5.1 tables.
2. Identify all constraints (views, summary tables, triggers, SQL functions, SQL
   methods, referential constraints) that point from the new tables to the
   Commerce Suite 5.1 tables or from Commerce Suite 5.1 tables to the new tables.
3. Drop all those constraints. If there are any SQL errors due to constraints in the
   migratedb.log after running the database migration script, you will need to
   drop the constraints before migrating database.

To drop the constraints, run the SQL statements as described in the sections below.

DB2 Databases

For DB2 databases, run the SQL statement as follows:

1. Open a command prompt window.
2. If you are running WebSphere Application Server as the root user, switch to the
   DB2 instance user ID for your database (for example, db2inst1):

   ```
   su - db2_instance_user_id
   ```

   If you are running WebSphere Application Server as the non-root user, switch
   to the WebSphere Application Server user ID (for example, wasuser):

```
su - WAS_user
```
3. Connect to your Commerce Suite database as a user with administrative authority for the database:
```
db2 connect to db_name user db_user_name using db_password
```
4. Enter the following SQL statement:
```
db2 ALTER TABLE table_name DROP constraint constraint_name
```

where:
  - *table_name* is the name of the customized table containing the constraint.
  - *constraint_name* is the name of the referential integrity constraint you wish to drop.
5. Note the constraints that you drop, as you will need to them after the data migration.

Oracle databases

For Oracle databases, run the SQL statement as follows:
1. Open a command window.
2. Switch to the Oracle user ID (for example, `oracle`):
```
su - oracle_user_ID
```
3. At the prompt, type:
```
sqlplus userid/password@service name
```

where:
  - *userid* is your Oracle user ID
  - *password* is the password for the Oracle user ID
  - *service name* is the Oracle service name

For example:
```
sqlplus oracle/oracle@o809
```
4. Enter the following SQL statement:
```
ALTER TABLE table_name DROP constraint constraint_name;
```

where:
  - *table_name* is the name of the customized table containing the constraint.
  - *constraint_name* is the name of the referential integrity constraint you wish to drop.
5. Note the constraints that you drop, as you will need to them after the data migration.

Migrating your database schema

After you have eliminated all items flagged by the database preparation script, you can run the following database migration script to update your schema to the WebSphere Commerce 5.4 level. For an overview of what the migration script does, see Appendix C, "Overview of the migration script," on page 159. For a summary of the database schema changes between Commerce Suite 5.1 and WebSphere Commerce 5.4, see Appendix E, "Database schema changes," on page 171.

Note that you can only run the migration script once on the database.

Note: For DB2 databases running on UNIX® systems, databases are normally cataloged as R*data_base_name* by default. For example, MALL will be cataloged as RMALL by default). To avoid problems when running the database migration script for remote databases, you need to do the following:

1. In the migratedb.sh script, update the following line:

   ```
   db2 connect to $database
   ```

   to:

   ```
   db2 connect to $database user $user using $password
   ```

2. When you run migratedb.sh, the database name should be R*data_base_name*, for example, RMALL

__ 1. If you are using DB2, open a command prompt window.

__ 2. If you are using DB2 and running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, db2inst1):

   ```
   su - db2_instance_user_id
   ```

   If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

   ```
   su - WAS_user
   ```

__ 3. If you are using Oracle and running WebSphere Application Server as the root user, ensure you are logged in as the root user.

   If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

   ```
   su - WAS_user
   ```

__ 4. Switch to the bin subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

__ 5. Run the database migration script as follows:

   ```
   ./migratedb.sh db_type db_name db_userID db_userID_password
   backup_dir atp | noatp [db2_install_dir] [db2_inst_home]
   [host_name] [port_number] [schema_owner]
   ```

   where:

   - *db_type* is the database management system you are using. The value can be either db2 or oracle.
   - *db_name* is the Commerce Suite 5.1 database that you are migrating (for example, mall).
   - *db_userID* is the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, mydbuser).
   - *db_userID_password* is the password for the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, mypasswd).
   - *backup_dir* is the directory where the migration script stores a backup copy of your database. For example, /my_backup. If you are running WebSphere Application Server as the non-root user (wasuser), ensure that this user has read and write access to the backup directory.
   - atp specifies that you wish to migrate your existing Commerce Suite 5.1 inventory data to the WebSphere Commerce 5.4 available-to-promise (ATP) inventory representation.

noatp specifies that you do not wish to migrate your existing Commerce Suite 5.1 inventory data to the WebSphere Commerce 5.4 available-to-promise (ATP) inventory representation.

For a summary of the new ATP inventory representation system that will help you decide whether you wish to migrate or not, please refer to "Migrating to ATP inventory" on page 5. If you migrate your inventory data to ATP representation, you will need to change your customized code related to inventory data. For more information, see the WebSphere Commerce 5.4 online help. If you decide to migrate your inventory data at a later time, you need to run the migrateATP script as described in Appendix D, "Converting to ATP inventory at a later time," on page 167.

Note that when you migrate to WebSphere Commerce 5.4 specifying atp, the database migration script does not populate any vendors in the RECEIPT table during the migration as there be multiple vendors per store or there may be none. If you do not have a vendor assigned to your store, you should create one using the WebSphere Commerce Accelerator before you migrate to the ATP inventory representation.

- *db2_install_dir* is the installation path for DB2. For example, /usr/lpp/db2_07_01. This parameter is required for DB2 databases only.
- *db2_inst_home* is the DB2 instance path for your DB2 instance. For example, /home/db2inst1/sqllib. This parameter is required for DB2 databases only.
- *host_name* is the fully qualified host name of your machine. **This parameter is required for Oracle databases only.**
- *port_number* is the Oracle listener port number. Typically, this is port 1521 by default. **This parameter is required for Oracle databases only.**
- *schema_owner* is the database schema owner user ID. You must enter the *schema_owner* in uppercase characters.

For example, to run the database migration script for a DB2 database, and convert your inventory representation to ATP, you can use the following command:

```
./migratedb.sh db2 mall mydbuser mypasswd /backup atp /usr/lpp/db2_07_01
  /home/db2inst1/sqllib DB2INST1
```

For example, to run the database migration script for an Oracle database and convert your inventory representation to ATP, you can use the following command:

```
./migratedb.sh oracle mall mydbuser mypasswd /backup atp
  myhost.montreal.ca 1521 WCS
```

__ 6. Check the migratedb.log file.

This log file is generated in the /usr/WebSphere/CommerceServer/logs/migration directory.

Ensure that no errors have occurred in running the script before proceeding. Search for the error or fail string in the log file. Also, you should check for SQLSTATE if you are using DB2, or ORA if you are using Oracle, when searching the log file. If any of these strings are present, you need to analyze the message, fix your data, and then reattempt the data migration.

You can ignore the following SQLSTATE messages if they appear in the migratedb.log:

```
SQLSTATE=23505  (duplicate entries)
```

```
================================================================================
```

```
LINE    MESSAGES FOR db2clpnc.bnd
------  -------------------------------------------------------------------
  SQL0595W  Isolation level "NC" has been escalated to "UR".
  SQLSTATE=01526

LINE    MESSAGES FOR db2arxnc.bnd
------  -------------------------------------------------------------------
  SQL0595W  Isolation level "NC" has been escalated to "UR".
  SQLSTATE=01526

================================================================================

SQL0100W  No row was found for FETCH, UPDATE or DELETE; or the result of a
query is an empty table.  SQLSTATE=02000

================================================================================
--------Warning------------
Message: [IBM][CLI Driver][DB2/NT] SQL0513W  The SQL statement will modify an
entire table or view.  SQLSTATE=01504

SQLState: 1504
Error Code: 513

================================================================================
Executed: update store set ALLOCATIONGOODFOR = 0
--------Warning------------
Message: [IBM][CLI Driver][DB2/NT] SQL0513W  The SQL statement will modify an
entire table or view.  SQLSTATE=01504

SQLState: 1504
Error Code: 513
```

You can also ignore the following error messages if they appear in the
database migration log. The reason is that your database has not loaded the
summary tables when you populate the database. The default is to populate
in the WebSphere Commerce 5.4 database with the wcs.summary.sql script.

```
DROP TABLE category

DB21034E  The command was processed as an SQL statement because it was not a

valid Command Line Processor command.  During SQL processing it returned:

SQL0204N  "DB2ADMIN.CATEGORY" is an undefined name.  SQLSTATE=42704


DROP TABLE product

DB21034E  The command was processed as an SQL statement because it was not a

valid Command Line Processor command.  During SQL processing it returned:

SQL0204N  "DB2ADMIN.PRODUCT" is an undefined name.  SQLSTATE=42704


DROP TABLE richattr

DB21034E  The command was processed as an SQL statement because it was not a

valid Command Line Processor command.  During SQL processing it returned:

SQL0204N  "DB2ADMIN.RICHATTR" is an undefined name.  SQLSTATE=42704


DROP TABLE richattrcatgp

DB21034E  The command was processed as an SQL statement because it was not a

valid Command Line Processor command.  During SQL processing it returned:

SQL0204N  "DB2ADMIN.RICHATTRCATGP" is an undefined name.  SQLSTATE=42704
```

```
DROP TABLE storeinv

DB21034E  The command was processed as an SQL statement because it was not a

valid Command Line Processor command.  During SQL processing it returned:

SQL0204N  "DB2ADMIN.STOREINV" is an undefined name.  SQLSTATE=42704
```

Database migration log and trace files

The database migration script generates various log files in the
/usr/WebSphere/CommerceServer/logs/migration directory.

The log files that are generated are:

migratedbscripts.log
> The log file for generating the schema scripts

migratedb.log
> The log file for the overall data migration including populating bootstrap
> data.

OrigSchema.log
> The detailed list of WebSphere Commerce Suite 5.1 tables for the from
> release in the database. For example, if you are migrating from WebSphere
> Commerce Suite database level 5.1.0.1, this log will contain a listing of all
> 5.1.0.1 WebSphere Commerce Suite schema tables.

TargetSchema.log
> The detailed list of WebSphere Commerce 5.4 schema tables in the
> database after successfully running the migration script. It includes unique
> indexes, column definitions, and constraints. Differences between the
> TargetSchema.log and OrigSchema.log files will yield any customization
> (for example, additional columns or tables you may have added) that you
> may have performed on the original shipped Commerce Suite tables.

The database migration script also generates the following log files in the
/usr/WebSphere/CommerceServer/logs directory:

messages.txt
> Contains the mass loader messages.

trace.txt
> Contains the trace file for the Mass Loader and ID Resolver. Note that the
> size of this file is usually very large.

ecmsg_*xxxxx*
> Contains the run time traces which are generated after you start the
> WebSphere Commerce server and start running the system.

To enable trace information (ecmsg_*xxxxx* files) see "Enabling tracing information"
on page 219.

Restoring custom constraints

After migrating your Commerce Suite 5.1 database, you need to restore any
referential constraints that you dropped in "Dropping custom constraints" on page
80.

Run the SQL statements as described in the sections below.

DB2 Databases

For DB2 databases, run the SQL statement as follows:

1. Open a command prompt window.
2. Switch to the DB2 instance user ID for your database (for example, db2inst1):

   `su - db2_instance_user_id`

3. Connect to your Commerce Suite database as a user with administrative authority for the database:

   `db2 connect to db_name user db_user_name using db_password`

4. Enter the following SQL statement:

   ```
   db2 ALTER TABLE table_name
      ADD CONSTRAINT constraint_name FOREIGN KEY (column_name)
      REFERENCES foreign_table_name ON DELETE CASCADE
   ```

   where:

   - *table_name* is the name of the customized table where you need to restore the constraint.
   - *constraint_name* is the name of the referential integrity constraint you wish to restore.
   - *column_name* is the name of the column on which the referential constraint is being applied.
   - *foreign_table_name* is the foreign table for which the referential constraint is being applied.

For examples of the SQL statements that add referential constraints, see the file `wcs.referential.sql` in the `/usr/WebSphere/CommerceServer/schema/db2` directory.

Oracle databases

For Oracle databases, run the SQL statement as follows:

1. Open a command window.
2. Switch to the Oracle user ID (for example, oracle):

   `su - oracle_user_ID`

3. At the prompt, type:

   `sqlplus userid/password@service name`

   where:

   - *userid* is your Oracle user ID
   - *password* is the password for the Oracle user ID
   - *service name* is the Oracle service name

   For example:

   `sqlplus oracle/oracle@o809`

4. Enter the following SQL statement:

   ```
   ALTER TABLE table_name
      ADD (CONSTRAINT constraint_name FOREIGN KEY (column_name)
      REFERENCES foreign_table_name ON DELETE CASCADE);
   ```

   where:

   - *table_name* is the name of the customized table where you need to restore the constraint.

- *constraint_name* is the name of the referential integrity constraint you wish to restore.
- *column_name* is the name of the column on which the referential constraint is being applied.
- *foreign_table_name* is the foreign table for which the referential constraint is being applied.

For examples of the SQL statements that add referential constraints, see the file wcs.referential.sql in the /usr/WebSphere/CommerceServer/schema/oracle directory.

Updating distinguished names

WebSphere Commerce 5.4 requires that the distinguished name (DN) column be populated for the ORGENTITY and USERS tables. You can use the migrateDN script to fill these tables with the recommended values except for guest users (type G). This script calls the fillorgDN.sql script which populates the DN column in the ORGENTITY table, and then populates the user's DN column in the USERS table. The fillorgDN.sql script is generated when you run the database migration script.

Before running the migrateDN script, do the following:

- Locate and view the contents of the fillorgDN.sql script to see the values for the DN column, and the values to which they will be updated. The fillorgDN.sql script is located in the schema subdirectory, for example /usr/WebSphere/CommerceServer/schema.
- If you do not wish to make any of the suggested updates, you will have to appropriately customize the script.

  Note: If you are using an LDAP server, you may need to edit the generated fillorgDN.sql to ensure the distinguished names (DN) of the organizational entities are what you want. When the organizational entities are created on the LDAP server later on by WebSphere Commerce, the DN values in the ORGENTITY table are used. For example, if you would like the organizational entity named Root Organization to be created under an LDAP entry with a DN value of c=US, you would change the DN for the Root Organization in fillorgDN.sql from o=Root Organization to o=Root Organization,c=US. Distinguished names for other organizational entities will also have to be modified accordingly. For instance, change the DN entry o=*YourOrganization*,o=Root Organization to o=*YourOrganization*,o=Root Organization,c=US.

Once you are satisfied with the content of the fillorgDN.sqlscript, run migrateDN as described in the section below.

DB2 databases

To run the migrateDN script for DB2 databases:

__ 1. Open a command prompt window.

__ 2. If you are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, db2inst1):

    su - *db2_instance_user_id*

If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

    su - *WAS_user*

   __ 3. Switch to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

   __ 4. Run the consistency checker on the migrated database as follows.

```
./migrateDN.db2.sh db_name db_userID db_userID_password db2insthome
```

where:

- *db_name* is the Commerce Suite 5.1 database that has been migrated to the WebSphere Commerce 5.4 database schema level (for example, `mall`).
- *db_userID* is the user ID to connect to the migrated database (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the migrated database (for example, `mypasswd`).
- *db2insthome* is the home directory for your DB2 instance user ID (for example, `/home/db2inst1/sqllib`)

For example, to run the script on the migrated `mall` database, you can use the following command:

```
./migrateDN.db2.sh mall mydbuser mypasswd /home/db2inst1/sqllib
```

   __ 5. Check the `migrateDN.log` file which is generated in the `/usr/WebSphere/CommerceServer/logs/migration` directory. Ensure that no errors have occurred in using the script before proceeding.

Oracle databases

To run the `migrateDN` script for Oracle databases:

   __ 1. Open a command window.

   __ 2. If you are running WebSphere Application Server as the root user, ensure you are logged in as the root user.

       If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

```
su - WAS_user
```

   __ 3. Switch to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

   __ 4. Run the consistency checker on the migrated database as follows.

```
./migrateDN.oracle.sh db_name db_userID db_userID_password
  host_name port_number
```

where:

- *db_name* is the Commerce Suite 5.1 database that has been migrated to the WebSphere Commerce 5.4 database schema level (for example, `mall`).
- *db_userID* is the user ID to connect to the migrated database (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the migrated database (for example, `mypasswd`).
- *host_name* is the fully qualified host name of your machine.
- *port_number* is the Oracle listener port number. Typically, this is port 1521 by default.

For example, to run the script on the migrated `mall` database, you can use the following command:

```
./migrateDN.oracle.sh mall mydbuser mypasswd myhost.montreal.ca 1521
```

__ 5. Check the `migrateDN.log` file which is generated in the
   `/usr/WebSphere/CommerceServer/logs/migration` directory. Ensure that no
   errors have occurred in using the script before proceeding.

Assigning a master catalog

WebSphere Commerce 5.4 requires that you have a master catalog designated for
each of your stores. To assign a master catalog for your store, you can run the
`choosemc.sql` script that is generated when you run the database migration script.
The migration script detects if you have more than one catalog in the database. If
you have only one catalog in the database, the migration script assigns this catalog
as the master catalog. In this situation, you do not need to follow the steps below.

Prior to running the `choosemc.sql` script, you will need to edit it. Locate and
replace the string `MASTERCATALOG_ID` with the reference number (primary key)
corresponding to the catalog that you wish to designate as your master catalog.
For example, locate the following statements in the script:

```
--please replace MASTERCATALOG_ID with one of the catalog of the store you want to
designate as MasterCatalog
update storecat set mastercatalog='1' where catalog_id=MASTERCATALOG_ID
    and storeent_id=10001;
insert into catgrptpc (catgroup_id,catalog_id,tradeposcn_id)
    values (0,MASTERCATALOG_ID,10006);
```

To choose catalog ID 6000 as your master catalog, you would update the statement
as follows:

```
update storecat set mastercatalog='1' where catalog_id=6000
    and storeent_id=10001;
insert into catgrptpc (catgroup_id,catalog_id,tradeposcn_id)
    values (0,6000,10006);
```

The following is an example of the output after you run the script:

```
-store :10001 has 20 catalogs.
--catalog:311000
--catalog:321000
--catalog:341000
--catalog:6000
--catalog:361000
--catalog:371000
--catalog:322000
--catalog:391000
--catalog:411000
--catalog:501000
--catalog:501000
--catalog:501000
--catalog:501000
--catalog:501000
--catalog:501000
--catalog:501000
--catalog:611000
--catalog:612000
--catalog:10001
--please replace MASTERCATALOG_ID with one of the catalog of the store you want to
designate as MasterCatalog
update storecat set mastercatalog='1' where catalog_id=6000
    and storeent_id=10001;
insert into catgrptpc (catgroup_id,catalog_id,tradeposcn_id)
    values (0,6000,10006);
```

To run the script refer to the following sections.

DB2 Databases

To run the `choosemc.sql` script for DB2 databases:

1. Open a command prompt window.
2. Switch to the DB2 instance user ID for your database (for example, `db2inst1`):

   `su - db2_instance_user_id`
3. Switch to the /usr/WebSphere/CommerceServer/schema directory.
4. Connect to your Commerce Suite database.

   `db2 connect to db_name user db_user_name using db_password`
5. Run the `choosemc.sql` script as follows:

   `db2 -tvf choosemc.sql`

Oracle databases

To run the `choosemc.sql` script for Oracle databases:

1. Open a command window.
2. Switch to the Oracle user ID (for example, `oracle`):

   `su - oracle_user_ID`
3. Switch to the /usr/WebSphere/CommerceServer/schema directory.
4. Enter the following command:

   `sqlplus -s userid/password@service_name < choosemc.sql`

   where:
   - *userid* is your Oracle user ID
   - *password* is the password for the Oracle user ID
   - *service_name* is the Oracle service name.

   For example:

   `sqlplus -s oracle/oracle@o809 < choosemc.sql`

Changing the status of orders and order items

It is recommended, but not required that you change all your orders and order items that have a status of `C` to a status of `S` in order for you to work with orders and order items using the Commerce Accelerator tools. The considerations you need to make are described in "Orders and order items" on page 7. To change the status, you can use the `ctos.sql` script that is generated when you run the database migration script.

To run the script, refer to the following sections.

DB2 Databases

To run the `ctos.sql` script for DB2 databases:

1. Open a command prompt window.
2. Switch to the DB2 instance user ID for your database (for example, `db2inst1`):

   `su - db2_instance_user_id`
3. Switch to the /usr/WebSphere/CommerceServer/schema directory.
4. Connect to your Commerce Suite database.

   `db2 connect to db_name user db_user_name using db_password`
5. Run the `ctos.sql` script as follows:

   `db2 -tvf ctos.sql`

Oracle databases

To run the `ctos.sql` script for Oracle databases:

1. Open a command window.
2. Switch to the Oracle user ID (for example, `oracle`):

   `su - oracle_user_ID`
3. Switch to the /usr/WebSphere/CommerceServer/schema directory.
4. Enter the following command:

   `sqlplus -s userid/password@service_name < ctos.sql`

   where:
   - *userid* is your Oracle user ID
   - *password* is the password for the Oracle user ID
   - *service_name* is the Oracle service name.

   For example:

   `sqlplus -s oracle/oracle@o809 < ctos.sql`

Bootstrap data for default stores

If you have made any changes to the bootstrap data for your default store (identified by `STOREENT_ID` set to 0) before migration, you need to make the equivalent changes to the bootstrap data for the default store, after you migrate to WebSphere Commerce 5.4. For the default store, the data migration process loads the default bootstrap data and this will overwrite your customizations you made to the bootstrap data. For non-default stores, the data migration process leaves the bootstrap data for the store unchanged.

For example, suppose before migration, you have modified your implementation of the `OrderProcessCmd` command to `OrderProcessBonusImpl` in the `CMDREG` table. After running the database migration script, the entry is reset to the default value of `OrderProcessImpl`. In this case, you will need to manually modify it back to `OrderProcessBonusImpl`. Otherwise, when you access the `OrderProcessCmd` command, it returns a blank page. With this change, you can access the page correctly.

Note: It is normally recommended that you do not modify any of the bootstrap files. If modified, an instance may not be created successfully.

WebSphere Payment Manager consideration

In WebSphere Commerce Suite 5.1, in the bootstrap data for the default store, the default implementation class for the `DoPaymentCmd` command is `DoPaymentCmdImpl`. This implementation does not use WebSphere Payment Manager.

But in WebSphere Commerce Business Edition 5.1 or WebSphere Commerce 5.4, the default implementation class for `DoPaymentCmd` is `DoPaymentMPFCmdImpl`. This implementation uses WebSphere Payment Manager.

If your WebSphere Commerce Suite 5.1 stores use the default implementation class `DoPaymentCmdImpl`, then after migration to WebSphere Commerce 5.4, if you still want to use `DoPaymentCmdImpl`, you can do one of the following:

1. Specify the `STOREENT_ID` in the `CMDREG` table if you still want to use `DoPaymentCmdImpl` for this store. This method is preferred as you do not modify the bootstrap data for WebSphere Commerce 5.4 and it is better for migrating

to the later version. For example, insert a new entry in `CMDREG` specifying the store's `STOREENT_ID=0` with `interfacename=DoPaymentCmd` and `classname=DoPaymentCmdImpl`.

2. Modify the bootstrap data in the `CMDREG` table. For default stores with the `STOREENT_ID=0`, change the default implementation class for `DoPaymentCmd` to `DoPaymentCmdImpl`. This is not recommended as you are modifying the bootstrap data for the default store.

For additional WebSphere Payment Manager considerations, see Chapter 9, "Payment component migration considerations," on page 143.

Verifying the database migration

This section provides some guidance for verifying that you have successfully migrated your database.

Viewing the migratedb.log file

After running the migration script, you should review the content of the `migratedb.log` file. Search for the string, `Migrating`, and check to see if `Total errors=0` is present. Note that you can ignore any warning. For a successful migration, your `migratedb.log` file should appear similar to the following:

```
Migration starts... Date:2001-10-02
Migrating Member table...
 ...
 ...
 ...
Summary
Total changed =
Total inserted =
Total queries =
Total warnings = 0
Total errors = 0


Migrating Catalog Entries..
...
...
...
Summary
Total changed =
Total inserted =
Total queries =
Total warnings =
Total errors = 0


Migrating Inventory..
...
...
Summary
Total changed =
Total inserted =
Total queries =
Total warnings = 0
Total errors = 0


Migrating Discount Data...
...
...
```

```
Summary
Total changed =
Total inserted =
Total queries =
Total warnings = 0
Total errors = 0



Migrating Contract Component..
...
...
Summary
Total changed =
Total inserted =
Total queries =
Total warnings =0
Total errors = 0



Migrating Calculation Framework..
...
...
Summary
Total changed =
Total inserted =
Total queries =
Total warnings =0
Total errors = 0.
```

Additional Checks

After checking the database migration log file, you should run the following SQL
statements against your database.

- Check the FLOW table by running the following select statement:

  ```
  DB2 SELECT * FROM FLOW
  ```

  If the flow table is empty, then that means the Mass Loader has problems
  loading business flow data. You should view the migratedb.log and message.txt
  files for more information.

- Check the distinguished names column, DN, in the ORGENTITY and USERS tables by
  running the following select statements:

  ```
  DB2 SELECT DN FROM ORGENTITY
  ```

  ```
  DB2 SELECT DN FROM USERS
  ```

  If any of the DN entries are empty, you likely did not run the migrateDN script for
  your database after migration, as described in "Updating distinguished names"
  on page 87.

Running the database consistency checker

After you have successfully run the generated migration script for your database,
run the consistency checker to check the state of the migrated database.

The checker generates a report and lists the items described in "Database
consistency checker output" on page 95.

Note: The consistency checker tool checks to see if any known integrity problems
are present in the migrated database. Even if you run the consistency

checker without receiving any error message, this does not necessarily guarantee that integrity issues are not present in the database. The tool may not cover all possible cases, especially for customized databases.

DB2 databases

To run the consistency checker for DB2 databases:

__ 1. Open a command prompt window.

__ 2. If you are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, db2inst1):

```
su - db2_instance_user_id
```

If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

```
su - WAS_user
```

__ 3. Switch to the bin subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

__ 4. Run the consistency checker on the migrated database as follows.

```
./dbchecker.db2.sh db_name db_userID db_userID_password db2insthome
```

where:

- *db_name* is the Commerce Suite 5.1 database that has been migrated to the WebSphere Commerce 5.4 database schema level (for example, mall).
- *db_userID* is the user ID to connect to the migrated database (for example, mydbuser).
- *db_userID_password* is the password for the user ID to connect to the migrated database (for example, mypasswd).
- *db2insthome* is the home directory for your DB2 instance user ID (for example, /home/db2inst1/sqllib).

For example, to run the consistency checker on the migrated mall database, you can use the following command:

```
./dbchecker.db2.sh mall mydbuser mypasswd /home/db2inst1/sqllib
```

__ 5. Check the dbchecker.log which is generated in the /usr/WebSphere/CommerceServer/logs/migration directory. Ensure that no errors have occurred in using the script before proceeding.

Oracle databases

To run the consistency checker for Oracle databases:

__ 1. Open a command window.

__ 2. If you are running WebSphere Application Server as the root user, ensure you are logged in as the root user.

If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

```
su - WAS_user
```

__ 3. Switch to the bin subdirectory where WebSphere Commerce 5.4 is installed. For example /usr/WebSphere/CommerceServer/bin.

__ 4. Run the consistency checker on the migrated database as follows.

```
./dbchecker.oracle.sh db_name db_userID db_userID_password
  host_name port_number
```

where:

- *db_name* is the Commerce Suite 5.1 database that has been migrated to the WebSphere Commerce 5.4 database schema level (for example, `mall`).
- *db_userID* is the user ID to connect to the migrated database (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the migrated database (for example, `mypasswd`).
- *host_name* is the fully qualified host name of your machine.
- *port_number* is the Oracle listener port number. Typically, this is port 1521 by default.

For example, to run the consistency checker on the migrated `mall` database, you can use the following command:

`./dbchecker.oracle.sh mall mydbuser mypasswd myhost.montreal.ca 1521`

__ 5. Check the `dbchecker.log` which is generated in the `/usr/WebSphere/CommerceServer/logs/migration` directory. Ensure that no errors have occurred in using the script before proceeding.

Database consistency checker output

If the consistency checker generates the following return codes, you must take the *mandatory* actions that are listed before proceeding with the migration process for your system. Otherwise, the WebSphere Commerce 5.4 runtime will not function with the migrated data.

Return Code	Explanation/Action
401	Your migrated data contains items that do not have a parent product. You must assign a parent product for each of these items. To create a parent product for an item, add an entry to the `CATENTREL` table.
402	Your migrated data contains items that have more than one parent product. You must ensure that only one parent product is assigned to each item. To delete additional parent products for items that are flagged as having more than one parent product, remove entries from the `CATENTREL` table.
405	The migrated `USER` table contains organizational inconsistencies. You need to correct any items that are flagged. For a user with a record in the `BUSPROF` table, if the `ORG_ID` and the `ORGUNIT_ID` columns are not null, starting with `ORGUNIT_ID`, use the `ORGENTITY` table to traverse up the organization hierarchy until `MEMBER_ID` eventually contains null or contains the same value as `ORGENTITY_ID`. The value of the `MEMBER_ID` column in `ORGENTITY` is not the same as the `ORG_ID` column in `BUSPROF`. Make sure that the value of the `MEMBER_ID` column in `ORGENTITY` is the same as the `ORG_ID` column in `BUSPROF`.
500	None of your catalogs are designated as a master catalog. Ensure that one of your catalogs are designated as a master catalog if you have multiple catalogs per store. Design a master catalog, and choose a master catalog by running the `choosemc.sql` script as documented in "Assigning a master catalog" on page 89.
503	Your migrated data does not contain any top-level catalog groups. You must ensure that your data contains at least one top-level

catalog group for each master catalog. Add the top-level catalog group relationship by adding an entry to the CATTOGRP table.

If the consistency checker generates the following return codes, it is recommended but not mandatory that you take the *optional* actions that are listed before proceeding with the migration process for your system. Otherwise, you will not be able to use the WebSphere Commerce 5.4 tools such as the Product Management tool on the migrated data.

Return Code	Explanation/Action
408	Your migrated data contains catalog groups that are under more than one trading position container. You need to ensure that your catalog groups are not under more than one trading position container. Delete any additional trading position container relationships by removing entries from the CATGRPTPC table.
501	Your migrated data contains catalog groups with more than one parent catalog group. Ensure that each of your catalog groups only designate one catalog group as its parent. Delete any additional parent catalog groups by removing entries from the CATGRPREL table.
502	Your migrated data contains catalog entries belonging to more than one catalog group. Ensure that every catalog entry belongs to only one catalog group. Delete any additional catalog groups by removing entries from the CATGPENREL table.
614	In the ATTRVALUE table, there needs to be a row for each attribute value defined with a CatEntryId = 0. For example, if you have two items that have the color red, the ATTRVALUE table should look like:

CATENTRY_ID	ATTRBUTE_ID	STRINGVALUE
0	color_id	red
item1_id	color_id	red
item2_id	color_id	red

Note that this is not a complete table. With this method, even if there are not any items defined, you can display valid attribute values that are defined for attributes. If multiple languages exist then each row needs to be defined again for each language.

This is step is required only if you want to use the WebSphere Commerce 5.4 tools such as the Product Management tool to manage your catalogs.

Add one row to the ATTRVALUE table with CATENTRY_ID set to 0 (zero) for each case where more than one item has the same ATTRBUTE_ID.

The catalogs for the Commerce Suite 5.1 sample stores, InFashion or WebFashion, do not have well-formed catalog trees that are suitable for a WebSphere Commerce 5.4 master catalog, You need to redesign the catalog according to the requirements of a master catalog as described in "Important structural considerations" on page 8, otherwise, the WebSphere Commerce 5.4 catalog editor tool, Product Management, may run into problems when navigating the catalog. For examples, of well-formed catalog trees, see the sample stores supplied with WebSphere Commerce 5.4 in the /usr/WebSphere/CommerceServer/samples directory.

Resetting LANGUAGE_ID

If you have set `LANGUAGE_ID` in your `STORE` table for your default store as described in "Setting up the STORE table and populatedb script" on page 71, you need to reset back to its original value after you have completed the database migration. For example, if `LANGUAGE_ID` was originally set to `-1` (`en_US`) before you modified it, you need to set it back as follows before continuing with other migration steps:

```
DB2 UPDATE STORE SET LANGUAGE_ID=-1 WHERE STORE_ID=0
```

Remigrating your database

If you have completed database migration and wish to remigrate the database for some reason, ensure that you remove all the files previously generated by the database migration script before you attempt to the remigrate your original database.

The following is the list of files to delete:
- For DB2 — under the `/usr/WebSphere/CommerceServer/schema/db2/migration` directory.

  For Oracle — under the `/usr/WebSphere/CommerceServer/schema/oracle/migration` directory.
  - `delwcs.constraint.sql`
  - `wcs.schema.create.sql`
  - `wcs.temp.drop.sql`
- Under the `/usr/WebSphere/CommerceServer/schema` directory:
  - `fillorgDN.sql`
  - `ctos.sql`
  - `choosemc.sql` (if present)
- Under the `/usr/WebSphere/CommerceServer/logs/migration` directory:
  - `migratedb.log`
  - `premigrate.log`
  - `migrateDN.log`
  - `migratedscripts.log`
  - `migrate.log`
  - `dbchecker.log`
  - `OrigSchema.log`
  - `TargetSchema.log`

When remigrating the database, your need to perform all the steps in this chapter including rerunning the database preparation script as described in "Running the database preparation script" on page 74.

Note: If you are remigrating your database after having completed the WebSphere Commerce instance migration and merchant key migration in Chapter 5, "Post data migration actions," on page 99, you do not have to complete those steps once again. Here are the steps to remigrate your database without remigrating the instance and merchant key:

1. Delete all the files listed above before attempting to remigrate your database.
2. Backup your migrated WebSphere Commerce 5.4 database.

3. Export the migrated USERREG table in your WebSphere Commerce 5.4 database to a text file. This table will contain the migrated passwords for users of your WebSphere Commerce system.

4. Restore your backed up Commerce Suite 5.1 database.

5. Perform all the steps in this chapter to migrate the restored Commerce Suite 5.1 database to the WebSphere Commerce 5.4 level.

6. Import the USERREG table from the text file in step 3. This table will contain the migrated passwords for users of your WebSphere Commerce system.

Chapter 5. Post data migration actions

The following sections describe migration actions you need to take after you have migrated your data to the WebSphere Commerce 5.4 level. It includes the following:

- "Migrating your Commerce Suite 5.1 instance configuration"
- "Migrating your security configuration" on page 104
- "Migrating store file assets" on page 109
- "Updating custom code for discount, shipping, or taxation" on page 115

Migrating your Commerce Suite 5.1 instance configuration

This section describes how to migrate your Commerce Suite 5.1 instance configuration to a WebSphere Commerce 5.4 instance configuration. For information on how to create a new WebSphere Commerce 5.1 instance, refer to the section "Create or modify an instance with Configuration Manager" in the *WebSphere Commerce Suite 5.1 Installation Guide* for your platform.

Steps before migrating your instance configuration

CAUTION:
You must migrate your database schema to WebSphere Commerce 5.4 before migrating your instance.

Before migrating your WebSphere Commerce Suite 5.1 instance to WebSphere Commerce 5.4:

- Ensure that the `LANG` environment variable correctly on your system for your national language. To see the value of the `LANG` environment variable, from a command prompt, enter the following command:

  ```
  echo $LANG
  ```

  If it is not set correctly, you need to add the following statement in your `.profile` file:

  ```
  export LANG=locale
  ```

  where *locale* is the locale corresponding to your national language, for example, en_US for English (United States). The `LANG` environment variable must be set correctly in order for the instance migration to work properly.

- WebSphere Commerce Suite 5.1 used UTF-8 as the default encoding for all languages in the *instance_name*.xml file. For WebSphere Commerce 5.4, the *instance_name*.xml file requires language specific encoding. Check the encoding for your *instance_name*.xml file and ensure that it is set correctly for your national language. For example, edit your *instance_name*.xml file located in /usr/WebSphere/CommerceServer/instances/*your_instance*/xml and search for:

  ```
  <?xml version="1.0" encoding="xxxxxx"?>
  ```

  In the above string, `encoding` should be set as follows:

Language	Encoding
Japanese	Shift_JIS

Language	Encoding
Korean	euc-Kr
Simplified Chinese	GB2312
Traditional Chinese	Big5
All other languages (including English)	ISO-8859–1

For example, if your national language is Japanese, set the encoding as:

```
<?xml version="1.0" encoding="Shift_JIS"?>
```

- Ensure that you have enabled SSL as described in "Enable SSL" on page 58.
- If you have manually backed up your Commerce Suite 5.1 customized files and directories, you must now copy these files and directories from your back up locations to the appropriate WebSphere Commerce 5.4 locations.

  If you are following the procedures in "Minimizing downtime during migration" on page 15, you also need to restore the instance configuration files in addition to the above files.

  To restore the files and directories from the back up locations, do the following:

  1. Open a command window and navigate to your temporary backup directory.
  2. Select the appropriate files or directories and copy them to the appropriate WebSphere Commerce 5.4 directory.

- Manually copy the following items. If your WebSphere Commerce Suite 5.1 system is on a machine remote from your WebSphere Commerce 5.4 machine, you need to copy the following items from your WebSphere Commerce Suite 5.1 machine to your WebSphere Commerce 5.4 machine.

  – The WebSphere Commerce Suite 5.1 `wcs_instances` file from /usr/lpp/CommerceSuite/instances to the new /usr/WebSphere/CommerceServer/instances directory.

  – The `/usr/lpp/CommerceSuite/instances/`*your_instance* directory (including its subdirectories) to the new /usr/WebSphere/CommerceServer/instances/*your_instance* directory.

  – The file `cfg.passwd` from /usr/lpp/CommerceSuite/bin to the new /usr/WebSphere/CommerceServer/bin directory.

  If your WebSphere Commerce Suite 5.1 system is on a machine remote from your WebSphere Commerce 5.4 machine, modify your *instance_name*.xml (for example, demo.xml) and other related .xml files and change all occurrences of the old host name to new host name of the WebSphere Commerce 5.4 machine. The following is a list of the files that need to be updated:

  – *instance_name*.xml
  – *instance_name*_was.xml
  – *instance_name*_was_DB.xml
  – *instance_name*_was_EJB.xml
  – *instance_name*_was_Start.xml
  – *instance_name*_was_Stores.xml
  – *instance_name*_was_Tools.xml
  – *instance_name*_was_VH.xml

- Edit the `wcs_instances` file in the /usr/WebSphere/CommerceServer/instances directory. Locate and change any occurrences of /usr/lpp/CommerceSuite/instances/*your_instance* (the WebSphere Commerce

Suite 5.1 instance path name) to
/usr/WebSphere/CommerceServer/instances/*your_instance* (the WebSphere
Commerce 5.4 instance path name).

- You also need to make the following updates to your *instance_name*.xml file (for
  example, demo.xml) located in the
  /usr/WebSphere/CommerceServer/instances/*your_instance* directory prior to
  instance migration:
  - **If you are running WebSphere Application Server as the non-root user
    (wasuser)**, change the port attribute under the Websphere node from 900 to
    2222 in the *instance_name*.xml file.
  - In WebSphere Commerce 5.4, the WebSphere Commerce administration tools
    run on a separate port than the server. The default port to run these tools is
    8000. If you wish to override this setting and use a different port for the
    administration tools, locate the WebSphere node in the *instance_name*.xml file,
    and add the ToolsPort="*port_number*" attribute to its attribute list, and set the
    value of *port_number* to the non-default port number.
    CAUTION:
    **Since editing the *instance_name*.xml file is error-prone, you are strongly
    advised not to override the default port.**
- Remove any occurrence of *instance_name*.xml.bak in the
  /usr/WebSphere/CommerceServer/instances/*your_instance* directory.
- Before running the instMigration.sh utility, ensure that the user ID that runs
  the utility (that is, root or wasuser) has full permission to access files and
  subfolders in the /usr/WebSphere/CommerceServer/instances directory.

Attention: If WebSphere Application Server security is turned on, you must
disable it before creating your instance as follows:

1. Open the WebSphere Application Server Administration console.
2. Click **Console > Security Center** and clear the **Enable Security** check box on
   the General tab.
3. Click **Finish**.
4. Restart the WebSphere Application Server administrative server.

Migrating your instance configuration

Note that you will need to perform the following steps to migrate each Commerce
Suite instance you are running.

Note: If you wish to run the instMigration.sh script as the non-root user
(wasuser), you have to comment out the following lines near the beginning
of the script.

```
if [ `whoami` != "root" ] ; then
        echo "You must be root to run this."
        exit 1
fi
```

Without commenting these lines, the instance migration script will not run
for a non-root configuration.

To migrate your Commerce Suite 5.1 instance, do the following:

__ 1. Start WebSphere Application Server.
__ 2. If you are running the instance migration as the root user, ensure you are
      logged on as the root user.

If you are running the instance migration as the non-root user, switch to the WebSphere Application Server user ID (for example, wasuser):

`su - WAS_user`

__ 3. From a command prompt, switch to the /usr/WebSphere/CommerceServer/bin directory and run the `instMigration.sh` script.

__ 4. To verify that the instance migration completed successfully, check the `instMigrate.log` file in the /usr/WebSphere/CommerceServer/instances directory.

You should see a statement indicating that the migration completed successfully at the end of the log to indicate a successful migration of the instance.

Also, to check whether the Enterprise JavaBeans (EJBs) deployment is successful, check the `WASConfig.log` file (the log file for importing EJBs) in the /usr/WebSphere/CommerceServer/instances/*instance_name*/logs directory. Ensure that no exceptions have occurred.

If you have made any customizations to your Web server configuration in Commerce Suite 5.1, ensure that they are still present in your current Web server configuration files. Otherwise, you should reapply your customizations into the configuration files.

You will restart your Web server and the `WebSphere Commerce Server –` *instance_name* application server after you migrate your security configuration as described in the next section.

Turning off the rules service

The instance migration process enables the Commerce rules server by default even if you have disabled the rules server component in Commerce Suite 5.1. To disable the rules server in WebSphere Commerce 5.4 after instance migration, change the `enable` directive file from `true` to `false` in the *instance_name*.xml file. The file is located in the /usr/WebSphere/CommerceServer/instances/*your_instance* directory.

Locate the following lines, and update the `enable` directive from:

```
<component enable="true"
   name="Rule Services"
   compClassName="com.ibm.commerce.rules.RulesSystemComponentConfiguration">
```

to:

```
<component enable="false"
   name="Rule Services"
   compClassName="com.ibm.commerce.rules.RulesSystemComponentConfiguration">
```

Copying store assets to new location

When migrating to WebSphere Commerce 5.4, store assets need to be copied from several WebSphere Commerce Suite directories to WebSphere Application Server directories as listed in the following table. The assets, which comprise the web application for the store, need to be copied over due to the new web application structure required by WebSphere Application Server 4.0.2.

You need to manually copy the contents of the source directories to the corresponding target directories that are indicated in the table below.

Note: The source and target directories are constructed using the values of the entries under the `<devtools>` element in the *instance_name*.xml in the table below.

Table 5. Copying store assets to new location in WebSphere Commerce 5.4

Entries in *instance_name*.xml file	Source directory (in WebSphere Commerce Suite 5.1) (These directories are presented assuming the default installation path.)	Target Directory (in WebSphere Commerce 5.4) (These directories are presented assuming the default installation path.)
StoresDocRoot + StoreWebPath	/usr/lpp/CommerceSuite/stores/web	/usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear/wcstores.war
StoresDocRoot + StoresPropertiesPath	/usr/lpp/CommerceSuite/stores/properties	/usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear/wcstores.war/WEB-INF/classes

In order to prevent the possibility of unauthorized retrieval of a store archive file from the store's URL to view all items that were used to load the store, ensure that all *store_name*.sar files on your machine get moved to the `/usr/WebSphere/CommerceServer/instances/`*your_instance*`/sar` directory. If the `sar` subdirectory is not present in your migrated system, you need to create it manually. When you create a new store in WebSphere Commerce 5.4 using Store Services, the tool creates the `sar` subdirectory for you.

Updating your WebSphere Commerce instance

In order to publish a migrated store, such as Infashion or WebFashion, modify the `TempPath` attribute within the `DevTools` node in your WebSphere Commerce instance file, from:

```
TempPath="/opt/WebSphere/CommerceServer/tools/devtools/temp"
```

to:

```
TempPath="/usr/WebSphere/CommerceServer/temp/your_instance/tools/devtools"
```

The *your_instance*.xml file is located in the `/usr/WebSphere/CommerceServer/instances/`*your_instance*`/xml` directory.

Running the install_was script

Before restarting the WebSphere Commerce server, you need to run the `install_was.sh` script that is provided with the WebSphere Commerce 5.4 FixPak in order to minimize any problems with the migration.

Before running the script, shut down any Web servers that are running as well as WebSphere Application Server

To run the script, go to the directory where you have unpackaged the files provided with the WebSphere Commerce 5.4 FixPak and enter:

```
./install_was.sh WAS_install_path WC_application_path
```

where:

- *WAS_install_path* is the WebSphere Application Server installation path (for example, /usr/WebSphere/AppServer).
- *WC_application_path* is the WebSphere Application Server installed applications path for your WebSphere Commerce instance (for example, /usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear).

Enter the paths when prompted.

Note that you need to run this script once for each WebSphere Commerce instance on your system. To see the instances that you have on your system, see the `wcs_instances` file in the /usr/WebSphere/CommerceServer/instances directory.

Restarting the Web server and WebSphere Application Server
After completing the instance migration, restart your Web server and WebSphere Application Server as follows:

1. Stop WebSphere Application Server.
2. Stop your Web server.
3. Restart your Web server.
4. Restart WebSphere Application Server
5. Restart the `WebSphere Commerce Server — instance_name` application server in the WebSphere Application Server console.

Enabling WebSphere Application Server EJB Security
If you had WebSphere Application Server EJB Security enabled in WebSphere Commerce 5.1, you need to re-enable it for WebSphere Commerce 5.4. See the section "Enabling WebSphere Application Server security" in the *WebSphere Commerce 5.4 Installation Guide* for the steps to redeploy the EJB security.

Note that it is highly recommended that that your machine meets the following requirements:

- A minimum machine memory of 1 GB.
- A minimum heap size of 384 MB, for the WebSphere Commerce application.

Verifying successful instance migration
To verify that the instance is migrated successfully, check the `instMigrate.log` file in the /usr/WebSphere/CommerceServer/instances directory. In addition, verify that the WebSphere Commerce server can be started without errors. These errors, if they occur, will be captured in the `wcs.log` file in the /usr/WebSphere/CommerceServer/instances/*instance_name*/logs directory.

Migrating your security configuration

After you have migrated your Commerce Suite 5.1 instance configuration to the WebSphere Commerce 5.4 level, you are required to run the merchant key migration utility, `MigrateEncryptedInfo.sh`. This utility will:

- Change the merchant key of a specified instance to a new key, and update the encrypted data in the store database according to the associated configuration file. WebSphere Commerce 5.4 requires that you use a non-default merchant key. If you are using a default merchant key in Commerce Suite 5.1, you have to change this default key to a non-default key.
- Change the way that logon passwords are stored in the database to conform to the WebSphere Commerce 5.4 requirements.
- Based on the setting of the `PDIEncrypt` flag in the *instance*.xml file, all the unencrypted credit card data will be encrypted, or all the encrypted credit card data will be unencrypted.

  The encrypted data that gets updated includes encrypted passwords and credit card data. The utility updates encrypted data in the following tables:
  - USERREG
  - PATTRVALUE
  - ORDPAYINFO
  - ORDPAYMTHD

Java virtual machine heap size consideration

The `MigrateEncryptedInfo` script sets the maximum Java virtual machine heap size (the maximum size of memory a JVM can use for its heap in megabytes) to be 512 MB. If you notice that the script is running out of JVM heap memory, you can edit the script and increase the two occurrences of the `-Xmx512m` setting. For example, to increase the maximum heap size to 1024 MB, you would change both occurrences of this setting to `-Xmx1024m`. The maximum allowable value depends on how much physical memory your database server machine has and how much memory is used by other applications. Usually, setting the JVM maximum heap size to one-third of your physical memory is a reasonable target. There is no value to setting this parameter to less than 64m (that is the default value).

Confirm encryption setting

Before running the merchant key migration utility, confirm that the PDI encryption setting in the *your_instance*.xml file for your WebSphere Commerce instance is set correctly. You can find the *your_instance*.xml file in the /usr/WebSphere/CommerceServer/instances/*your_instance*/xml directory. Edit the *your_instance*.xml file (for example, demo.xml), search for the string, `PDIEncrypt`, and set its value to either ″on″ or ″off″.

PDIEncrypt=″on″
> Sensitive information such as credit card data is stored in encrypted format using the merchant key.

PDIEncrypt=″off″
> Sensitive information such as credit card data is stored in plain text format (unencrypted).

DB2 Databases

Before running the `MigrateEncryptedInfo.sh` utility, ensure that the user ID that runs the utility (that is, `db2inst1` or `wasuser`) has full permission to access files and subfolders in the /usr/WebSphere/CommerceServer/instances/*your_instance* directory.

Run the `MigrateEncryptedInfo` utility as follows:

__ 1. Stop the `WebSphere Commerce Server — instance_name` application server in the WebSphere Application Server console.

__ 2. If you are running `MigrateEncryptedInfo.sh` using a non-database user ID (for example, `wasuser`) you must first run the `db2profile` script:

> `. db2_instance_user_home/sqllib/db2profile`

> where *db2_instance_user_home* is the home directory for your DB2 instance user ID (for example, /home/db2inst1).

> For example, the `db2profile` script can be run as follows:
> `. /home/db2inst1/sqllib/db2profile`

__ 3. Open a command prompt window.

__ 4. If you are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, `db2inst1`):

> `su - db2_instance_user_id`

> If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

> `su - WAS_user`

__ 5. Traverse to the `bin` subdirectory under the WebSphere Commerce 5.4 installation directory. For example:

`/usr/WebSphere/CommerceServer/bin`

__ 6. Run the `MigrateEncryptedInfo` utility for DB2 as follows:

```
./MigrateEncryptedInfo.sh  db2
  [instance_name] [current_key] [new_key]
```

where:

- *instance_name* is the name of the instance which will be updated. This parameter can be left out if there is only one instance installed.
- *current_key* is the current merchant key, in clear text (ASCII) form. You should specify this parameter only if you are currently using a non-default Commerce Suite 5.1 merchant key. If you are using the default Commerce Suite 5.1 merchant key, do not specify this parameter as the utility will detect that you are using the default key.
- *new_key* is the new merchant key, in clear text (ASCII) form. It must conform to the following rules:
  - Its length must be 16 hexadecimal characters. Characters can be one of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, or f.
  - It must contain a minimum of one alphabetic character.
  - It must contain a minimum of one numeric character.
  - It must be in lowercase.
  - It cannot contain more than 4 consecutive characters.

  For example, `aaaa1aaaa1aaaa12` and `abcdeaaaa3aaaa12` are allowed but `aaaaabaaaa1aaaa1` is not allowed.

  For examples of how to specify the command in different scenarios, see the sections below.

__ 7. Restart your Web server and WebSphere Application Server:
  a. Stop WebSphere Application Server.
  b. Stop your Web server.
  c. Restart your Web server.
  d. Restart WebSphere Application Server

__ 8. Start the `WebSphere Commerce Server – instance_name` application server in the WebSphere Application Server console.

The tool generates two log files in the `/usr/WebSphere/CommerceServer/logs` directory:

- `CCInfoMigration.log`
- `MKChangeUserAndCCInfoMigration.log`
- `MigrateEncryptedInfoError.log`

You should review the information in these log files and ensure they do not contain any error messages. Errors similar to the following can be ignored, if and only if, there is some existing data that was not encrypted with the specified *current_key*. If your data has been encrypted using different keys, you will need run the `MigrateEncryptedInfo` script multiple times, specifying a different *current_key* each time. In this case, only a subset of the encrypted data would be migrated during each pass, so there would be several of these errors in the log which you can ignore. However, the presence of such errors could also indicate a different problem, in which case the messages should not be ignored.

Unintentionally specifying the incorrect key may result in these error messages, and customers might not be able to log on. :

```
%3DES-F-CIPHERINIT; Exception caught while initializing the cipher object.
; java.lang.ArrayIndexOutOfBoundsException: 16

%3DES-F-DCRYPT; Exception caught while decrypting
; javax.crypto.IllegalBlockSizeException: Input length (with padding)
  not multiple of 8 bytes
```

Note that you can use this utility to update your merchant key only at migration time. If you wish to change your merchant key at a later time after migration, use the Configuration Manager to update the key. See the WebSphere Commerce 5.4 online help sections on using Configuration Manager for more information.

Examples for a DB2 database

In the following examples, the instance name `myinstance` is only required if you have multiple Commerce Suite 5.1 instances.

- If you are using the default Commerce Suite 5.1 merchant key, and have an instance called `myinstance`, specify only the new key:

  `./MigrateEncryptedInfo.sh  db2 myinstance abcdef0123456789`

- If you are using a non-default Commerce Suite 5.1 merchant key, that is, you have already changed your merchant key to be unique, and you wish to change the key for WebSphere Commerce 5.4, specify both the current key and the new key:

  `./MigrateEncryptedInfo.sh  db2 myinstance 0123456789abcdef`
  `    abcdef0123456789`

- If you are using a non-default Commerce Suite 5.1 merchant key, that is, you have already changed your merchant key to be unique, and you *do not* wish to update the key for WebSphere Commerce 5.4, you do not specify either key:

  `./MigrateEncryptedInfo.sh  db2 myinstance`

  (Note that if you specify the current key and the new key with same value, you will receive an error message to that effect.)

Oracle Databases

Run the `MigrateEncryptedInfo` utility as follows:

__ 1. Stop the `WebSphere Commerce Server — instance_name` application server in the WebSphere Application Server console.

__ 2. Open a command window.

__ 3. If you are running WebSphere Application Server as the root user, ensure you are logged on as the root user.

   If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

   `su - WAS_user`

__ 4. Traverse to the `bin` subdirectory under the WebSphere Commerce 5.4 installation directory. For example:

   `/usr/WebSphere/CommerceServer/bin`

__ 5. Run the `MigrateEncryptedInfo` utility for Oracle databases as follows:

   `./MigrateEncryptedInfo.sh oracle [instance_name] [current_key] [new_key]`

   where:

   - *instance_name* is the name of the instance which will be updated. This parameter can be left out if there is only one instance installed.

- *current_key* is the current merchant key, in clear text (ASCII) form. You should specify this parameter only if you are currently using a non-default Commerce Suite 5.1 merchant key. If you are using the default Commerce Suite 5.1 merchant key, do not specify this parameter as the utility will detect that you are using the default key.
- *new_key* is the new merchant key, in clear text (ASCII) form. It must conform to the following rules:
  - Its length must be 16 hexadecimal characters. Characters can be one of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, or f.
  - It must contain a minimum of one alphabetic character.
  - It must contain a minimum of one numeric character.
  - It must be in lowercase.
  - It cannot contain the same character repeated more than 4 times in a row.

  For example, `aaaa1aaaa1aaaa12` and `abcdeaaaa3aaaa12` are allowed but `aaaaabaaaa1aaaa1` is not allowed.

  For examples of how to specify the command in different scenarios, see the sections below.

__ 6. Restart your Web server and WebSphere Application Server:
  a. Stop WebSphere Application Server.
  b. Stop your Web server.
  c. Restart your Web server.
  d. Restart WebSphere Application Server

__ 7. Start the `WebSphere Commerce Server` – *instance_name* application server in the WebSphere Application Server console.

The tool generates two log files in the `/usr/WebSphere/CommerceServer/logs` directory:

- `CCInfoMigration.log`
- `MKChangeUserAndCCInfoMigration.log`
- `MigrateEncryptedInfoError.log`

You should review the information in these log files and ensure they do not contain any error messages. Errors similar to the following can be ignored, if and only if, there is some existing data that was not encrypted with the specified *current_key*. If your data has been encrypted using different keys, you will need run the `MigrateEncryptedInfo` script multiple times, specifying a different *current_key* each time. In this case, only a subset of the encrypted data would be migrated during each pass, so there would be several of these errors in the log which you can ignore. However, the presence of such errors could also indicate a different problem, in which case the messages should not be ignored. Unintentionally specifying the incorrect key may result in these error messages, and customers might not be able to log on. :

```
%3DES-F-CIPHERINIT; Exception caught while initializing the cipher object.
; java.lang.ArrayIndexOutOfBoundsException: 16

%3DES-F-DCRYPT; Exception caught while decrypting
; javax.crypto.IllegalBlockSizeException: Input length (with padding)
  not multiple of 8 bytes
```

Note that you can use this utility to update your merchant key only at migration time. If you wish to change your merchant key at a later time after migration, use

the Configuration Manager to update the key. See the WebSphere Commerce 5.4 online help sections on using Configuration Manager for more information.

Examples for an Oracle database

In the following examples, the instance name `myinstance` is only required if you have multiple Commerce Suite 5.1 instances.

- If you are using the default Commerce Suite 5.1 merchant key, and have an instance called `myinstance`, specify only the new key:

  `./MigrateEncryptedInfo.sh oracle myinstance abcdef0123456789`

- If you are using a non-default Commerce Suite 5.1 merchant key, that is, you have already changed your merchant key to be unique, and you wish to update the key for WebSphere Commerce 5.4, specify both the current key and the new key:

  `./MigrateEncryptedInfo.sh oracle myinstance 0123456789abcdef abcdef0123456789`

- If you are using a non-default Commerce Suite 5.1 merchant key, that is, you have already changed your merchant key to be unique, and you *do not* wish to change the key for WebSphere Commerce 5.4, you do not specify either key:

  `MigrateEncryptedInfo.sh oracle myinstance`

  (Note that if you specify the current key and the new key to have the same value, you will receive an error message to that effect.)

Migrating store file assets

This section describes how to migrate a published Commerce Suite 5.1 store to WebSphere Commerce 5.4.

Before making your changes to your JavaServer Pages, ensure that you have:

- Published your Commerce Suite 5.1 store archives. See the section "Publish a store archive" in the Commerce Suite 5.1 online help for the detailed steps to publish store archives.
- Migrated your database schema as described in Chapter 4, "Migrating your database," on page 69.
- Migrated your instance to the WebSphere Commerce 5.4 level as described in "Migrating your Commerce Suite 5.1 instance configuration" on page 99.

This automatically migrates the store data in the database to WebSphere Commerce 5.4.

After you have completed the above, you need to create a new WebSphere Commerce 5.4 store archive based on the sample stores provided. See the section "Creating a store" and "Creating a store archive" in the WebSphere Commerce 5.4 online help. Your old Commerce Suite 5.1 store archive will no longer work with the Store Services tools in WebSphere Commerce 5.4.

> **Important**
>
> - Migrated stores will not have a `index.jsp`, which is new for WebSphere Commerce 5.4. Consequently, the WebSphere Commerce Store Services launch store button will not work for migrated stores. In order to launch your store using the Store Services launch store button, you need to create your own `index.jsp`. See the sample store provided with WebSphere Commerce 5.4 for an example of an `index.jsp` file and refer to the WebSphere Commerce online help describing the supplied sample stores.
> - You can launch your store with the Web address you used in Commerce Suite 5.1, prior to migration. For example:
>
>   ```
>   http://hostname/webapp/wcs/stores/servlet/StoreCatalogDisplay?
>       storeId=storeId&langId=-1&catalogId=catalogId
>   ```
>
>   where *storeId* is the store ID number for your store and *catalogId* is the catalog number. See "Publishing a store archive using the command line&" in the WebSphere Commerce online help for more information.
> - If you have customized code or commands written in Java or Enterprise JavaBeans objects in WebSphere Commerce 5.1 or WebSphere Commerce Business Edition 5.1, you need to redeploy your them to the level required for WebSphere Commerce 5.4. See the "Transitioning your customized code" section of the *WebSphere Commerce Studio 5.4 Migration Guide*. This transition is required due to the move to WebSphere Application Server 4.0.2 from IBM WebSphere Application Server 3.5.
>
>   If you have customized some EJBs, confirm the JNDI names are the same as they are called in customized commands after migration.
>
>   If any of your JSPs or customized code previously used the `com.ibm.util` package from IBM WebSphere Application Server 3.5.x, you have to rewrite your code or JSP by using the equivalent class from the IBM SDK for Java that is packaged with WebSphere Application Server 4.0.2. The `com.ibm.util` package has been removed from the current version of WebSphere Application Server.

Updating the Commerce Suite 5.1 JavaServer Pages

There are several changes you need to make to your Commerce Suite 5.1 JavaServer Pages (JSPs) in order to make them operational for WebSphere Commerce 5.4. A tool to automatically do this, `migrateJSP.sh` is provided. To run the tool:

1. Open a command window.
2. If you are running WebSphere Application Server as the root user, run the script as the root user.

   If you are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

   ```
   su - WAS_user
   ```
3. Switch to the /usr/WebSphere/CommerceServer/bin directory.
4. Run the `migrateJSP` script as follows:

   ```
   ./migrateJSP.sh inputDir outputDir
   ```

   where

- *inputDir* is the fully-qualified name of the directory where your Commerce Suite 5.1 JSP resides.
- *outputDir* is the fully-qualified name of the directory where the converted WebSphere Commerce 5.4 JSP will reside.

For example:

```
./migrateJSP.sh /my_wcs_jsp_dir /my_wcbe_jsp_dir
```

Note: You should not run this tool more than once on your JavaServer Pages or it may result in syntax errors in your JSPs. For example, the first time you run the script, your JSP will be correctly updated: `<jsp:include page="<%=incfile%>"/>` updated to `<jsp:include page="<%=incfile%>" flush="true"/>`. If you run it once more, it gets updated to `<jsp:include page="<%=incfile%>" flush="true" flush="true/>`

The following are the required changes that the above tool will automatically make to your JSPs. This list is derived from migrating the InFashion store from Commerce Suite 5.1 to WebSphere Commerce 5.4.

- As WebSphere Commerce 5.1 uses the JavaServer Page 1.1 level, you need to change all occurrences of the following:

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

  to

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

  The JavaServer Page templates used in stores running in Commerce Suite 5.1 were required to support the JavaServer Page 1.0 specification. When migrating your store to WebSphere Commerce 5.4, you must ensure that the JavaServer Page templates in your store comply to the JavaServer Page 1.1 specification created by Sun Microsystems. For information about the JavaServer Page 1.1 specification, refer to Sun Microsystems' Java web site at `http://java.sun.com`.

- Change all occurrences of the following link (if it exists):

  ```
  <Form NAME="BillAddressForm" METHOD="POST" action="<%="OrderCopy"%>">
  ```

  or

  ```
  <Form NAME=BillAddressForm METHOD="POST" action="<%="OrderCopy"%>">
  ```

  to

  ```
  <Form NAME="BillAddressForm" METHOD="POST" action="OrderCopy">
  ```

Following is a list of additional changes you can make to your Commerce Suite 5.1 JSPs. Note that these changes are optional:

- You can replace some Commerce Suite 5.1 commands with the current WebSphere Commerce 5.4 version of the command. Note, that these changes are not mandatory, but are recommended to be completed at some point in time.
  - You can replace all occurrences of the `getCalculatedPrice()` method with the `getCalculatedContractPrice()` method. These methods occur in the `ItemDataBean` and `ProductDataBean` classes. For more information on these methods and classes, see the WebSphere Commerce 5.4 online help.
  - You can update the usage of the `CatalogEntryMPE` bean to the `EMarketingSpot` bean. For example, in the Commerce Suite 5.1 `newarrivals.jsp` for the InFashion store, the `CatalogEntryMPE` bean was used as follows:

```
<%
// create the e-Marketing Spot
CatalogEntryMPE productSpot = new CatalogEntryMPE();

//LOOK: Set the right spot name.
productSpot.setName("NewArrivalsPage");
productSpot.setMaximumNumberOfItems(new Integer(20));

//Set the default list of promoted products to the
//contents of the HOMEPAGE_PROMO category.
List defaultCatalogEntryIdList = new ArrayList();
if (newArrivalCategoryId != null )
{
        CategoryDataBean subCategories[];
%>
```

The following is an example of the same function in the WebSphere Commerce 5.4 `newarrivals.jsp` for the WebFashion store:

```
<!-- START PROMO -->
<%
// create the e-Marketing Spot
EMarketingSpot eMarketingSpot = new EMarketingSpot();

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

// 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();
%>
```

– You can use `UsablePaymentTCListDataBean` instead of `ProfileCassetteAccountDataBean` and `UserRegistrationDataBean.findUser()` instead of `UserRegistrationDataBean.getRegisterType()` to check the user type.

To see how to create a new WebSphere Commerce 5.4 store, see the section "Creating a store" in the WebSphere Commerce 5.4 online help.

Changes to shipaddress.jsp

If you wish to launch your migrated store through the WebSphere Commerce 5.4 Store Services and have your shoppers be able to complete the shopping flow in your store, you need to make the following update to the `shipaddress.jsp` that was shipped in Commerce Suite 5.1.

Locate and change the following line from the Commerce Suite 5.1 `shipaddress.jsp`:

```
if ( !addr.getAddress1().equals("-"))
```

to:

```
 if (addr.getAddress1()!=null && !addr.getAddress1().equals("-"))
```

You can update your `shipaddress.jsp` in the following directory:

`/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/store_dir`

where *store_dir* is the directory for your store, for example, `webfashion1`.

If you have used the code from the Commerce Suite 5.1 `shipaddress.jsp` in your own application, you will need to make the corresponding change.

Shoppers should launch the store from a clean browser (that is, newly launched) after shutting down all browsers used to run any of the tools such as Store Services.

Changes to register.jsp and account.jsp

The `register.jsp` and `account.jsp` JSPs shipped in Commerce Suite 5.1 for the sample InFashion store did not return any errors if there is a login failure to the store.

In order have stores return error codes when a customer fails to logon you need to update the `register.jsp` and `account.jsp`. In addition, you need to update the `infashiontext_en_US.properties` file.

For a listing of the updated JSPs and the property file see at:
- "register.jsp" on page 200
- "account.jsp" on page 205
- "infashiontext_en_US.properties" on page 209

The required changes to these JSPs for WebSphere Commerce 5.4 relative to the original files shipped with Commerce Suite 5.1 are indicated by bold font.

If you have used the code from the Commerce Suite 5.1 `register.jsp` and `account.jsp` from InFashion in your own applications, you will need to make the corresponding changes in your code.

User registration for sample WebFashion store

For the sample Commerce Suite 5.1 WebFashion store that is available as a download from the Commerce Suite 5.1 Web site, you need to make the following change to the `register.jsp` provided. Replace the command `RegisterNAddToMemberGroup` with the command `UserRegistrationAdd`. See the WebSphere Commerce online help for usage and syntax information for the `UserRegistrationAdd` command.

This change is necessary as the `RegisterNAddToMemberGroup` command hard codes the `owner_id` to a value of `0` (zero). In WebSphere Commerce 5.4, the `owner_id` is actually a value of `-2001`. Without this change, you will not be able to register a new user in WebSphere Commerce 5.4.

After making the above change to `register.jsp` file, you need to change the customer profile in order to obtain discounts for the registered user. Using the Commerce Accelerator, change the `register10` customer profile for the migrated WebFashion store as follows:

1. Log into WebSphere Commerce Accelerator and select the WebFashion store.
2. Select **Marketing > Customer Profile**.
3. Select **register10** and click **Change** on the right navigation bar.
4. The Change Customer Profile page displays. On the left navigation bar, select **Registration > Registration status**.
5. Set the registration status to **Registered**.
6. Click **OK** on the bottom right corner of the screen to change the `register10` customer profile.

With Commerce Suite 5.1, when a new user is registered in the Commerce Suite 5.1 WebFashion store, the `RegisterNAddToMemberGroup` command automatically assigns the user under the member group `register10`. Thus, all the registered users for the Commerce Suite 5.1 WebFashion store belong to the `register10` member group in the `MBRGRPMBR` table. After changing the `register.jsp` file, the new registered users in the migrated WebFashion store no longer belong to the `register10` member group.

If you have used the code from the Commerce Suite 5.1 WebFashion `register.jsp` in your own applications, you will need to make the corresponding changes in your code.

Change to iiDB.getDate() method in InterestItemDisplay.jsp

In the `wcstores.war\InterestItemDisplay.jsp` file, method `iiDB.getDate()` is not defined in the WebSphere Commerce 5.4 class, `com.ibm.commerce.catalog.beans.InterestItemDataBean`. WebSphere Commerce 5.4 uses the `getLastUpdate()` method to replace the WebSphere Commerce 5.1 `getDate()` method. Thus, in the `InterestItemDisplay.jsp` file, you should replace method `iiDB.getDate()` by `iiDB.getLastUpdate()`.

Updating store properties files

Migrated Commerce Suite 5.1 stores may be using a store-specific version of the following properties files:

- `Address.properties`, or `Address_locale.properties` for different national languages (for example, `Address_en_US.properties` for English)
- `UserRegistration.properties`, or `UserRegistration_locale.properties` for different national languages (for example, `UserRegistration_en_US.properties` for English)

For these stores, you need to remove these files so that the WebSphere Commerce 5.4 system's default properties are used. The format of the Address and UserRegistration properties have changed in WebSphere Commerce 5.4 and the WebSphere Commerce Accelerator tools only work with the new properties file.

You can update the two sets of properties files for the store, `UserRegistration` and `Address`, as follows:

1. Back up all `Address_*.properties` files (that is, property files beginning with `Address_`) that are located in the `/usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_instance_name.ear/wcstores.war/WEB- INF/classes/store_dir` directory.

2. Copy all `Address_*.properties` files from the `/usr/WebSphere/CommerceServer/samples/properties` directory to the `/usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_instance_name.ear/wcstores.war/WEB- INF/classes/store_dir` directory.

If you have any customized any `UserRegistration` or `Address` properties files, you need to convert them to the new format. For information on how to write the new versions of these properties, see the WebSphere Commerce 5.4 online help for class `PropertyResourceBundleReader`.

New sample store directory

WebSphere Commerce Suite 5.1 stored the store archive files for sample stores in the `samples` subdirectory (`/usr/WebSphere/CommerceServer/samples`). For the

WebSphere Commerce 5.4 Store Services tool to display the Commerce Suite 5.1 stores, you need to move the .sar files from the old `samples` subdirectory to the new `sample store` subdirectory.

Store status unpublished in Store Services

After migrating your stores, the status of the migrated store that was previously published in Commerce Suite 5.1 is displayed incorrectly as `not published` in the Store Services tool. When migrating to WebSphere Commerce 5.4, the database migration script clears all the entries which have SCSSTATE set to 'C' (complete) in the SCHSTATUS table. The database migration script moves all other entries in the SCHSTATUS table that are in a pending or incomplete state to the SCHACTIVE table. As the entries that were previously set to complete are not moved to the new SCHACTIVE table, your store's state will display as `not published` in Store Services after migration.

Note that your store will continue to function properly.

Turning off all caching

If you have caching enabled, you should disable it before publishing your store. You can turn off caching by updating your configuration in *instance*.xml using configuration manager.

If you do not disable caching, you may receive a `Transaction log for the database is full` message in the `message.txt` log. In this case, see the recommended solution to the problem in Appendix H, "Troubleshooting," on page 211.

Updating custom code for discount, shipping, or taxation

If you have any customized code for the discount, shipping, or taxation, you can use the following procedure to migrate your calculation code to WebSphere Commerce 5.4:

1. Identify whether there is any custom code for discount, shipping or tax.

   Check the CMDREG table using the following query:

   `SELECT * FROM CMDREG WHERE INTERFACENAME LIKE '%.ApplyOrder%Cmd'`

   If there is any customized code for the discount, shipping or taxation subsystems, the result set will not be null. Check the column CLASSNAME in the CMDREG table for the class names referenced by the commands.

2. Identify whether the customized code needs to be migrated.

   Check the custom implementations of the ApplyOrder*XXX*Cmd task commands to see if they extend or reference the Commerce Suite 5.1 default implementations, that is:

   - `ApplyOrderAdjustmentCmdImpl`
   - `ApplyOrderShippingChargesCmdImpl`
   - `ApplyOrderTaxesCmdImpl`
   - `ApplyCalculationUsagesCmdImpl` — the base implementation

   If they do not extend or reference the above implementations, theApplyOrder*XXX*Cmd task commands will continue to work in WebSphere Commerce 5.4. Note that migration of the task commands as described below is still recommended to maintain future compatibility.

3. Migrate the custom code:

   a. Identify the calculation usage ID of the custom code:

discount	-1
shipping	-2
sales tax	-3
shipping tax	-4

   b. Identify the corresponding default interface and implementation in WebSphere Commerce 5.4.

   Check the CALMETHOD table using the following query:

   ```
   SELECT * FROM CALMETHOD
       WHERE STOREENT_ID=-1 AND
       CALUSAGE_ID=calculationUsageID AND
       SUBCLASS=12
   ```

   Take note of the task command name, for example, `com.ibm.commerce.order.calculation.ApplyCalculationUsageCmd` for discount. The corresponding implementation is always the task command name appended with Impl, for example, `com.ibm.commerce.order.calculation.ApplyCalculationUsageCmdImpl` for discount.

   c. Create a new interface that extends the interface in step 3b. For example:

   ```
   package mypackage;
   import com.ibm.commerce.order.calculation.*;
   public interface MyApplyDiscountCmd extends ApplyCalculationUsageCmd {
           public static final String NAME = "mypackage.MyApplyDiscountCmd";
           public static final String_defaultCommandClassName = NAME + Impl;
   }
   ```

   Note that the `String_defaultCommandClassName` variable needs to match the name of the custom implementation, `mypackage.MyApplyDiscountCmdImpl`, in this example.

   d. Modify the custom implementation to extend the implementation in step 3b and to implement the interface in step 3c. For example:

   ```
   package mypackage;
   import com.ibm.commerce.order.calculation.*;
   public class MyApplyDiscountCmdImpl extends
      ApplyCalculationUsageCmdImpl implements MyApplyDiscountCmd {
        }
   ```

   e. Modify the custom implementation to match the WebSphere Commerce 5.4 interface signature.

   The `setItems()` method in WebSphere Commerce 5.4 is roughly equivalent to the `setOrderItems()` method in Commerce Suite 5.1 and there should be no need to override the default implementation of `setItems()`. Unlike Commerce Suite 5.1, the override is no longer necessary because in the `performExecute()` method, you can retrieve the list of (order) items by calling the protected method, `getItems()` .

   The `Item` class is a wrapper for the `OrderItemAccessBean` class:

   - To wrap an `OrderItemAccessBean` instance (or object):

     `Item item = new Item(abOrderItem);`

   - To unwrap an `OrderItemAccessBean` instance (or object):

     `OrderItemAccessBean abOrderItem = item.getOrderItem();`

   - It is recommended that your customized code interacts with `Item` instead of `OrderItemAccessBean`

- You need to commit changes to an `Item` so they can be picked up by the underlying `OrderItemAccessBean` and EJB cache:

  `item.commit();`

- If you choose to update the `OrderItemAccessBean` directly, call `item.refresh()` to make sure that the `Item` is in synchronized with the underlying `OrderItemAccessBean` instance (or object).

- The `setCurrency()` method in WebSphere Commerce 5.4 is equivalent to the `setOrderCurrency()` method in Commerce Suite 5.1. You can retrieve the (order) currency by calling the protected method, `getCurrency()`.

Check the WebSphere Commerce 5.4 online help for the above referenced classes and interfaces for more details.

4. Register the custom interface in the `CALMETHOD` table. The following `INSERT` statement is one example of how you can do this:

```
INSERT INTO CALMETHOD
  (CALMETHOD_ID, STOREENT_ID, CALUSAGE_ID, TASKNAME, DESCRIPTION, SUBCLASS, NAME)
VALUES
  (calculationMethodID, storeID, -1, 'interfaceName', 'my method to apply discount',
   12, 'custom ApplyCalculationUsage (discount)')
```

where:

- *calculationMethodID* is any positive number that does not conflict with another existing `calculationMethodId`, for example, 123456.

- *interfaceName* is the one from step 3c on page 116, for example, . `mypackage.MyApplyDiscountCmd`.

5. Update the `STENCALUSG` table to reference this calculation method. The following `INSERT` statement is one example of how you can do this:

```
INSERT INTO STENCALUSG
  (STOREENT_ID, CALUSAGE_ID, ACTCC_CALMETHOD_ID, ACTRC_CALMETHOD_ID,
   CALCODE_ID, CALMETHOD_ID_APP, CALMETHOD_ID_SUM, CALMETHOD_ID_FIN,
   USAGEFLAGS, CALMETHOD_ID_INI, SEQUENCE)
VALUES
  (storeID, -1, -1, -5, null, calculationMethodID, -203, null, 1, -201, 1.0)
```

where *calculationMethodID* is the one from step 4.

Verifying your migration

After you complete the above steps, perform the verification steps documented in the section "Verifying your installation" in the *WebSphere Commerce 5.4 Installation Guide* to ensure that all WebSphere Commerce 5.4 components are working correctly.

Chapter 6. Additional post data migration actions

The following sections describe migration actions you may decide to take depending on your requirements, after you have migrated your data to the WebSphere Commerce 5.4 level. It includes the following:

- "Migrating the rule server configuration"
- "Auctions" on page 124
- "Business accounts" on page 125
- "Reconfiguring the staging server for customized database tables" on page 125
- "Dropping obsolete Commerce Suite 5.1 tables" on page 126
- "Loading new cache triggers" on page 126
- "Retrofitting customized properties" on page 128
- "Shipping Calculation Codes" on page 128
- "Finding products and inventory" on page 128
- "Customized commands" on page 129
- "Product Advisor migration considerations" on page 129

Migrating the rule server configuration

You can skip this section if:

- You had no rule services configured under Commerce Suite 5.1.
- All your rule services were created by the Campaign tools. Campaign migration is already complete as a result of completing the previous migration steps in this Guide.

The file wcs.server, now located in the /usr/WebSphere/CommerceServer/instances/*your_instance*/xml/rules directory, was used to configure the Advisor Rule Server in Commerce Suite 5.1. In WebSphere Commerce 5.4, this configuration information is stored in the database.

To configure your rule services the same way they were configured in Commerce Suite 5.1, do the following:

1. Ensure that you have migrated your instance as documented in "Migrating your Commerce Suite 5.1 instance configuration" on page 99.
2. To view the wcs.server file, open it in a text editor. This is a file in XML format that is similar to the following figure:

```
<?xml version="1.0" ?>

<DeployRulesServerConfig>
    <Name>Stateless Event Poster server</Name>
    <ServerFactory>
        <JavaName>com.blazesoft.server.deploy.NdStatelessServer</JavaName>
    </ServerFactory>
    <ServiceManagerFactory>
        <JavaName>com.blazesoft.server.local.NdLocalServiceManager</JavaName>
    </ServiceManagerFactory>

    <DeployRulesServiceConfig>
        <Name>Loan Event Poster Argument Service</Name>
        <RulesServiceAgentFactoryFactory>
            <JavaName>com.blazesoft.server.rules.NdScriptRulesServiceAgentFactory</JavaName>
            <RulesProjectLoaderFactory>
                <JavaName>com.blazesoft.server.rules.NdRulesProjectFileLoader</JavaName>
                <Project>/usr/WebSphere/CommerceServer/instances/demo/rules/ConsumerCredit_POSTER.adv</Project>
            </RulesProjectLoaderFactory>
        </RulesServiceAgentFactoryFactory>
        <DeploymentType>Java</DeploymentType>
        <DeployRulesServiceClientContextFactory>
```

```
        <JavaName>com.blazesoft.server.deploy.rules.NdDeployPosterRulesServiceClientContext</JavaName>
        <SrlMappingClass>ScoredLoanApplication</SrlMappingClass>
    </DeployRulesServiceClientContextFactory>
    <NumAgents>2</NumAgents>
    <RecyclePolicy>0</RecyclePolicy>
</DeployRulesServiceConfig>

<DeployRulesServiceConfig>
    <Name>Loan Event Poster Argument Wrapping-Results Extractor Service</Name>
    <RulesServiceAgentFactoryFactory>
        <JavaName>com.blazesoft.server.rules.NdScriptRulesServiceAgentFactory</JavaName>
        <RulesProjectLoaderFactory>
            <JavaName>com.blazesoft.server.rules.NdRulesProjectFileLoader</JavaName>
            <Project>../../../data/rules/ConsumerCredit_POSTER.adv</Project>
        </RulesProjectLoaderFactory>
    </RulesServiceAgentFactoryFactory>
    <DeploymentType>Java</DeploymentType>
    <DeployRulesServiceClientContextFactory>
        <JavaName>com.blazesoft.server.deploy.rules.NdDeployPosterRulesServiceClientContext</JavaName>
        <SrlArgumentsObjectFactory>
            <SrlClass>ScoredLoanApplication</SrlClass>
            <SrlObjectInitializerFunctional>
                <SrlName>initServiceData</SrlName>
                <SrlArgumentType>string</SrlArgumentType>
            </SrlObjectInitializerFunctional>
        </SrlArgumentsObjectFactory>
        <SrlResultExtractorFunctional>
            <SrlName>extractServiceResult</SrlName>
        </SrlResultExtractorFunctional>
    </DeployRulesServiceClientContextFactory>
    <NumAgents>2</NumAgents>
    <RecyclePolicy>0</RecyclePolicy>
</DeployRulesServiceConfig>

</DeployRulesServerConfig>
```

The top-level tag is `DeployRulesServerConfig`. This entire tag represents a rule server. It contains a number of tags called `DeployRulesServiceConfig`. Each one of these tags represents a rule service. You can safely ignore the rule server tag (`DeployRulesServerConfig`). You will need to extract four pieces of information from each rule service tag (`DeployRulesServiceConfig`).

3. Launch the WebSphere Commerce 5.4 Administrator's Console.

4. Logon and select **store**. Choose the store name and click **OK**.

5. Navigate to the Rule Service Administration tool. (**Rule Services > Administration**).

6. For each `DeployRulesServiceConfig` tag, do the following:

   a. From the Rule Service Administration tool, select **Add Service**

   b. The system presents you with four input fields:

   Name of Rule Service
   Enter the value of the `<Name>...</Name>` tag. In this example, the first rule service's name is `Loan Event Poster Argument Service`.

   Project File Name
   Enter the value of the `<Project>...</Project>` tag. In this example, the first rule service's project path is:

   `/usr/WebSphere/CommerceServer/instances/demo/rules/ConsumerCredit_POSTER.adv</Project`

   Use forward slashes (/) to separate the different folder names and enter the fully-qualified path name of the `.adv` file for the rule project.

   You should have backed up your Commerce Suite 5.1 `.adv` files as described in "Backing up directories and files" on page 27

   Number of Agents
   Enter the value of the `<NumAgents>...</NumAgents>` tag. In our example, the first rule service's agents value is 2.

   Session Timeout
   Enter the value of the

`<ServiceSessionTimeout>...</ServiceSessionTimeout>` tag, if it exists. In this example, the first rule service's session time out value is not specified. In this case, you can use the default value of 30000, meaning 30000 milliseconds or 30 seconds.

  c. Click **OK**.

The system will show you the new list of rule services, which will include the rule service you have just added. Repeat these steps for each rule service in the file `wcs.server`.

7. When you have finished these steps, restart the WebSphere Commerce 5.4 server and return to the Rule Service Administration tool to verify that your rule services have been successfully migrated.

Rule server administration commands

Rule service administration commands have changed both behavior and interface for WebSphere Commerce 5.4. The Commerce Suite 5.1 versions of the commands, found in packages `com.ibm.commerce.rules.commands` and `com.ibm.commerce.ruleservice.admin.commands`, use the Scheduler to broadcast requests to all your application clones to add, change, remove or refresh rule services. For WebSphere Commerce 5.4, the commands have been replaced with better-named ones in the same packages. Also, the commands now operate in a *just-in-time* manner. For example, when you refresh a rule service, each application clone refreshes its own instance of that rule service as soon as it needs to execute the rule service again. This approach improves reliability and avoids unnecessary updates. If you have extended the rule service administration commands, you will need to examine the new commands to see how this change in behavior affects your customized extensions.

Table 6. Project and package level mapping of the Rules system

Project Name	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
WCS Rules Deployment	com.ibm.commerce.rules.deployment. commands	
WCS Rules Project Support	com.ibm.commerce.rules.project	com.ibm.commerce.rules.project
		com.ibm.commerce.rules.beans[1]
WCS Rules System	com.ibm.commerce.rules	com.ibm.commerce.rules
	com.ibm.commerce.rules.commands	com.ibm.commerce.rules.commands
	com.ibm.commerce.rules.beans[1]	
	com.ibm.commerce.rules.blaze[2]	
	com.ibm.commerce.rules.messages	
	com.ibm.commerce.rules.selector	
	com.ibm.commerce.rules.selector.blaze	
	com.ibm.commerce.rules.util[3]	
		com.ibm.commerce.rules.ecmessages
		com.ibm.commerce.rules.exception
WCS Rules System Admin	com.ibm.commerce.ruleservice.admin.beans	com.ibm.commerce.ruleservice.admin.beans
	com.ibm.commerce.ruleservice.admin. commands	com.ibm.commerce.ruleservice.admin. commands
	com.ibm.commerce.ruleservice.admin.util	com.ibm.commerce.ruleservice.admin.util
		com.ibm.commerce.ruleservice.admin.beansrc

Table 6. Project and package level mapping of the Rules system (continued)

Project Name	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
WCS Rules System Enterprise Beans	com.ibm.commerce.rules.objects	com.ibm.commerce.rules.objects
	com.ibm.commerce.rules.objimpl	com.ibm.commerce.rules.objimpl
	WCSMCRuleLayerEJBReserved	WCSMCRuleLayerEJBReserved
		WCSRuleServerEJBReserved
		com.ibm.commerce.rules.helpers
Rule Server Integration		com.ibm.commerce.rules.blaze [2]
		com.ibm.commerce.rules.blaze.exception
Rules Infrastructure		com.ibm.commerce.rules.repository
		com.ibm.commerce.rules.util[3]
Rules Infrastructure Common Services		com.ibm.commerce.services
		com.ibm.commerce.services.logging

Note: Even though a few packages are similar in WebSphere Commerce Suite 5.1 and WebSphere Commerce 5.4, they may have completely different classes. Similar classes may or may not have similar methods.

[1] [2] [3] These packages have moved to different projects in WebSphere Commerce 5.4.

The tables in the sections below list the changes to the Commerce Suite 5.1 Controller commands and application programing interfaces for rule server administration.

Controller commands

The behavior of the rule server administration controller commands have changed. Generally in WebSphere Commerce 5.4, they update rule service configuration information in the database, rather than broadcast information to all application clones. One exception is the following set of commands that require a broadcast:

`com.ibm.commerce.ruleservice.admin.commands.BroadcastUpdateRuleServiceStatusCommand`

These commands are used through corresponding URLs, and are not intended to be customized nor extended.

Refer to the WebSphere Commerce 5.4 online help for more information on any of these commands.

In the following table, only the base names of the commands are listed, for brevity. The `com.ibm.commerce.ruleservice.admin.commands.` portion of the complete command name is not included. For example, the complete name for the `AddRuleServiceCommand` command is `com.ibm.commerce.ruleservice.admin.commands.AddRuleServiceCommand`.

Table 7. Rule server controller commands

WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
AddRuleServiceCommand	AddRuleServiceCommand
Did not exist	BroadcastUpdateRuleServiceStatusCommand
StopRuleServiceCommand	DisableRuleServiceCommand
EditRuleServiceCommand	EditRuleServiceCommand
StartRuleServiceCommand	EnableRuleServiceCommand
RefreshRuleServiceCommand	RefreshRuleServiceCommand

Table 7. Rule server controller commands (continued)

WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
RemoveRuleServiceCommand	RemoveRuleServiceCommand
CheckRuleServiceStatusCommand	UpdateRuleServiceStatusCommand

Application programming interface calls (task commands)

What used to be direct method calls in Commerce Suite 5.1 are now task commands in WebSphere Commerce 5.4. The most commonly-used application programming interface (API) call in Commerce Suite 5.1 is to invoke a rule service. In WebSphere Commerce 5.4 this is done by using `com.ibm.commerce.rules.commands.InvokePersonalizationRuleServiceCommand`.

Refer to the WebSphere Commerce 5.4 online help for more information on any of these task commands.

In the following table, only the base names of the APIs and commands are listed, for brevity. For the Commerce Suite 5.1 commands, the `com.ibm.commerce.rules.RulesSystem.` portion of the complete API name is not listed. For example, the complete name for `changeServiceConfiguration()` is `com.ibm.commerce.rules.RulesSystem.changeServiceConfiguration()`. Similarly, for the WebSphere Commerce 5.4 commands, the `com.ibm.commerce.rules.commands.` portion of the command name is not listed. For example, the complete name for `ChangePersonalizationRuleServiceCommand` is `com.ibm.commerce.rules.commands.ChangePersonalizationRuleServiceCommand`.

Table 8. Rule server API calls (task commands)

WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
changeServiceConfiguration()	ChangePersonalizationRuleServiceCommand
addService()	CreatePersonalizationRuleServiceCommand
stopService()	DisablePersonalizationRuleServiceCommand
startService()	EnablePersonalizationRuleServiceCommand
invokeService()	InvokePersonalizationRuleServiceCommand
Did not exist	MarkPersonalizationRuleServiceChangedCommand
removeService()	RemovePersonalizationRuleServiceCommand
getService().getStatus()	UpdatePersonalizationRuleServiceStatusCommand

The following example illustrates implementation differences in API calls between WebSphere Commerce Suite 5.1 and WebSphere Commerce 5.4:

Invoking a rule service in Commerce Suite 5.1 (assuming the command context is set to `context` and the rule service name is `ruleServiceName`):

```
RuleServiceKey key=null;
key = new RuleServiceKey(ruleServiceName, context.getStoreId());
RulesSystem rulesSystem = RulesSystemToolbox.getInstance().getRulesSystem();
if (rulesSystem.isAvailable()) {
        rulesSystem.invokeService (key,context);
}
```

Invoking a rule service in WebSphere Commerce 5.4 (assuming the commandContext has been set to `commandContext` and the rule service name is `ruleServiceName`):

```
InvokePersonalizationRuleServiceCommand command =
    (InvokePersonalizationRuleServiceCommand)
CommandFactory.createCommand
    (InvokePersonalizationRuleServiceCommand.class.getName(), storeId);
command.setCommandContext(commandContext);
command.setServiceName(ruleServiceName);
command.execute();
```

Note that in the above example, the interface in WebSphere Commerce 5.4 is more straightforward and simpler to use. It hides many of the implementation details from a user.

Exception Handling

In WebSphere Commerce 5.4, the Rules System has extended exception handling to explicitly identify various problem areas. In WebSphere Commerce Suite 5.1, there was one `RulesExceptionHandler` class which handled all the generic Rules System related exceptions. In WebSphere Commerce 5.4, the `RulesExceptionHandler` class has been replaced by a number of Rules Exception classes. These distinct classes make it easier to identify the actual cause of the exception. These classes are contained in the `com.ibm.commerce.rules.exception` package in the WebSphere Commerce Rules System Project and they extend the general WebSphere Commerce RuntimeException class.

The Rules System exceptions are:
- InvalidRuleServiceKeyException
- PersonalizationRuleServerException
- PersonalizationRuleServerNotAvailableException
- PersonalizationRuleServiceExistsException
- PersonalizationRuleServiceNotEnabledException
- PersonalizationRuleServiceNotFoundException
- RuleServerConfigurationNotFoundException
- RuleServerNotFoundException
- RuleServiceNotFoundException
- RuleServiceConfigurationNotFoundException
- RulesSystemDataModelException
- RulesSystemRuntimeException

Auctions

If you have enabled auctions in Commerce Suite 5.1, you need to consider the following.

All auction order items in Commerce Suite 5.1 uses the default contract (`CONTRACT_ID` in the `CONTRACT` table). After migrating to WebSphere Commerce 5.4, the auction order items are asserted with the new default contract for WebSphere Commerce 5.4. The default contract in WebSphere Commerce 5.4 may have some terms and conditions and refund policies and so on which are not suitable for auction items.

You should review whether the default contract terms and conditions are what you require for auction order items. If they are not, you need to change the contract to one that is suitable for auctions by changing `TRADING_ID` in the `TRADING` table to

point to the suitable contract. By default, the database migration script sets TRADING_ID to point to the default contract that is created during the data migration.

Business accounts

The migration script does not create any business accounts. The script associates all migrated order items with the default contract. If you wish to create an account after migrating to WebSphere Commerce 5.4, you either have to specify to use the default contract or you need to modify your order items. For more information about creating an account, see the section "Creating a new business account" in the WebSphere Commerce 5.4 online help.

Reconfiguring the staging server for customized database tables

If you have customized database tables in Commerce Suite 5.1 and wish to use the Staging server in WebSphere Commerce 5.4, you need to run the Stage Copy utility command (stagingcopy) prior to the data migration to keep the product database and staging database consistent. After the data migration is complete, you need to reconfigure the Staging server. The migration process does not migrate the previous Commerce Suite 5.1 configuration.

Note that in the migration process, the Commerce Suite 5.1 staging server tables, are renamed by appending _WCS51 to the original name. Thus, the Commerce Suite 5.1 staging server tables are preserved as:

- STGSITETAB_WCS51
- STGMERTAB_WCS51
- STGMRSTTAB_WCS51
- STAGLOG_WCS51

You can view the contents of these renamed tables for reference.

Note that you may need to increase the log file size for the database. To increase the log file size, enter the following statement from a DB2 command window:

DB2 UPDATE DB CFG FOR *db_name* USING LOGFILSIZ *nnn*

where *db_name* is the name of your database, for example, MALL , and *nnn* is a number representing the LOGFILSIZ value. Refer to the DB2 administration information for more information.

In addition, the Commerce Suite 5.1 database cleanup utility table, is renamed by appending _WCS51 to the original name. Thus, the Commerce Suite 5.1 database cleanup utility table is preserved as CLEANCONF_WCS51. You can view the contents of this renamed table for reference.

To reconfigure your staging server for customized database tables, see the section "Configuring the staging server for customized tables" in the WebSphere Commerce 5.4 online help.

To reconfigure the database cleanup utility for customized database tables, see the section "Adding a new configuration to the Database Cleanup utility" in the WebSphere Commerce 5.4 online help.

Dropping obsolete Commerce Suite 5.1 tables

The following tables from Commerce Suite 5.1 are considered to be obsolete in WebSphere Commerce 5.4:

- CMPGNINTV
- CMPGNRV
- INTVMPE
- INTVSGMT
- MAFAMILY
- MATYPE
- MPE
- MPETYPE
- ONQUEUE
- ONLOG
- ONSLOG
- ORDERMSG
- SEGMENT
- ZIPCODE
- ACCCMDGRP
- ACCMBRGRP
- ACCCUSTEXC

Verify that you are no longer using the above tables. If you no longer require any of the tables, you can drop them as follows:

DB2   Type the following in a command window:

    db2 drop table *table_name*

where *table_name* is the name of your table you wish to drop.

Oracle Use the Database Configuration Assistant to delete the table you wish to drop.

Note: If there are entries in the KEYS or SUBKEYS table for any obsolete tables you have dropped, you need to remove those entries from the KEYS or SUBKEYS table.

Loading new cache triggers

Commerce Suite 5.1 provided cache triggers and the option for you to load them as well as unload them for performance reasons. Triggers are database events that occur when a row in a table is updated, added, or deleted. The cache component, which is enabled by default, uses triggers as a notification mechanism to indicate when an object is invalidated.

The migration script does not update your existing cache triggers. However, the following files are provided in the following directories:

- For DB2 databases — /usr/WebSphere/CommerceServer/schema/db2
- For Oracle databases — /usr/WebSphere/CommerceServer/schema/oracle

wcs.cache.trigger.sql
　　　Loads both the Commerce Suite 5.1 and WebSphere Commerce 5.4 cache triggers.

wcs.cache.trigger1.sql
　　　Loads only the Commerce Suite 5.1 cache triggers.

wcs.cache.trigger2.sql
　　　Loads only the WebSphere Commerce 5.4 cache triggers.

wcs.cache.delete.trigger.sql
　　　Unloads all Commerce Suite 5.1 and WebSphere Commerce 5.4 cache triggers.

wcs.cache.delete.trigger1.sql
　　　Unloads only the Commerce Suite 5.1 cache triggers.

wcs.cache.delete.trigger2.sql
　　　Unloads only the WebSphere Commerce 5.4 cache triggers.

If you have previously loaded the Commerce Suite 5.1 cache triggers, you need to:

1. Unload the Commerce Suite 5.1 triggers using `wcs.cache.delete.trigger1.sql` before database migration. This should have been already performed as described in "Dropping Commerce Suite 5.1 cache triggers" on page 79.
2. Load both the Commerce Suite 5.1 and WebSphere Commerce 5.4 triggers using `wcs.cache.trigger.sql`.

The steps are as follows:

For DB2 databases:
1. Open a DB2 command window.
2. Switch to the `/usr/WebSphere/CommerceServer/schema/db2` directory.
3. Connect to your Commerce Suite database:

   `db2 connect to db_name user db_user_name using db_password`
4. Load both the Commerce Suite 5.1 and WebSphere Commerce 5.4 triggers as follows:

   `db2 -tvf wcs.cache.trigger.sql`

For Oracle databases:
1. Open a command window.
2. Switch to the /usr/WebSphere/CommerceServer/schema/oracle directory.
3. Load both the Commerce Suite 5.1 and WebSphere Commerce 5.4 triggers as follows:

   `sqlplus userid/password@service_name < wcs.cache.trigger.sql`

where:
- *userid* is your Oracle user ID
- *password* is the password for the Oracle user ID
- *service_name* is the Oracle service name.

For example:

`sqlplus oracle/oracle@o809 < wcs.cache.triggers.sql`

Retrofitting customized properties

For any customized properties and code that you have backed up in "Premigration actions" on page 3, retrofit your customizations into the corresponding WebSphere Commerce 5.4 property file.

Shipping Calculation Codes

In Commerce Suite 5.1, shipping calculation codes were calculated separately for groups of order items with different shipping addresses. That is, shipping calculation codes grouped their order items by shipping address. Grouping by shipping address is now optional behavior. To ensure backward compatible behavior, the migration script sets the perAddress flag of the CALCODE.GROUPBY column during migration for all shipping calculation codes (those where CALCODE.CALUSAGE_ID = -2).

Finding products and inventory

After migration of your data, you need to perform the following steps to find products and inventory using the WebSphere Commerce Accelerator.

To find a product:

1. Logon to the WebSphere Commerce Accelerator and select the previously published store.
2. Click **Product** and then select **Find Product**
3. Enter the product code.

   For example, if sku-105 is an item belong to product sku-102, then you need to enter the product ID sku-102.
4. Enter 102, you can find this product,
5. Select SKU on the right panel, to find sku-105.

To find inventory:

1. Logon to the WebSphere Commerce Accelerator and select the published store.
2. Click **Logistics** and then select **Find Inventory**
3. Enter the SKU of the item in the field

   Notes:
   a. Enter the SKU of the item, not the SKU of the product. You can find the item's SKU from the Find Product field.
   b. Enter the number, for example, 105, then click **Find**, to find the item you want.

Note that all names and descriptions of the migrated items are set to "default migrated baseitem description". The name and description of these items are contained in the BASEITMDSC table. This table is new in WebSphere Commerce 5.4. Commerce Suite 5.1 did not have the names and descriptions for the items.

Customized commands

Consider if you have developed any customized commands in Commerce Suite 5.1, (for example, if you have created a controller command with an EJB). In this case, after you have migrated your system to WebSphere Commerce Business Edition 5.1, you need to redeploy all your customized commands as described in the *WebSphere Commerce Studio 5.4 Migration Guide* before running them in the migrated system.

Format for logonID in the edit_registration page

If you are using LDAP, the logon ID in the edit registration page will be shown in DN format rather than RDN format. In order to display it to RDN format, you can use a method that is provided in the `UserRegistrationDataBean` to properly retrieve the logon ID. In order to use this method you will need to make the following minor change in the JSP of each store. Note that the old method is still supported and so the migrated store will still work properly, if LDAP is not used.

Locate the following code in `edit_registration.jsp`:

```
<%
    strLogonID = jhelper.htmlTextEncoder(bnRegister.getLogonId());
  strPassword = bnRegister.getLogonPassword();
  strFirstName = jhelper.htmlTextEncoder(bnRegister.getFirstName());
  strLastName = jhelper.htmlTextEncoder(bnRegister.getLastName());
}

%>
```

Update the code as indicated in bold as follows:

```
<%
    // use getAttribute("RDN") here because getLogonId() will
    // return the DN value when LDAP is used
  strLogonID = jhelper.htmlTextEncoder(bnRegister.getAttribute("RDN"));
  strPassword = bnRegister.getLogonPassword();
  strFirstName = jhelper.htmlTextEncoder(bnRegister.getFirstName());
  strLastName = jhelper.htmlTextEncoder(bnRegister.getLastName());}

%>
```

Product Advisor migration considerations

If you are migrating your Product Advisor configuration from previous releases, observe the following items. It is assumed that you have created a WebSphere Commerce Suite 5.1 Product Advisor search space, and that Product Advisor is operational in your previous release of WebSphere Commerce.

1. The sample JavaServer Pages (JSPs) in WebSphere Commerce 5.4 (`pe51.jsp`, `pc51.jsp`, and `sa51.jsp` found in the `/usr/WebSphere/CommerceServer/samples/web/pa` directory) are the migrated versions of the files with the same names in Commerce Suite 5.1. For WebSphere Commerce 5.4, the datatype package names have changed as summarized in the table below. For any JSPs that reference these package names, you need to change occurrences of `com.ibm.commerce.datatype` to `com.ibm.commerce.**pa.**datatype` as summarized in the following table:

WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
com.ibm.commerce.datatype.DsString	com.ibm.commerce.**pa**.datatype.DsString
com.ibm.commerce.datatype.DsInteger	com.ibm.commerce.**pa**.datatype.DsInteger

WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
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

Note that there is also new data type introduced in WebSphere Commerce 5.4, `com.ibm.commerce.pa.datatype.DsLong`, that you should use for `catentry_id` or other attributes with values larger than a typical integer.

2. In your script to create a Product Advisor search space (for example, `createsearchspace.sh`) you also need to make the same datatype package name changes as indicated in the table in step 1 on page 129.

   You can refer to the sample `createsearchspace.sh` provided in WebSphere Commerce 5.4, which has been updated with the package name changes.

   Note that in WebSphere Commerce 5.4, a new method of search space creation is supported using the PAConfig utility. This utility helps automate the process for search space creation but requires an input XML file. You need to create the XML file manually. For more details on this process, refer to the topic, ″Product Advisor″, in the online help.

3. Run the script to create Product Advisor search spaces in your migrated WebSphere Commerce 5.4 system. After running the script, verify that all package names were migrated successfully by checking the `DATATYPE` column of the `ICEXPLFEAT` table. All class names in this column should have the new package name, `com.ibm.commerce.pa.datatype`.

4. Update the XML file used as input to the `PABatchXML` utility for creating metaphors, with the same package name changes listed in the table in step 1 on page 129.

5. Run the `PABatchXML` utility.

Part 2. Additional migration considerations

The chapters in this part of the Migration Guide describe additional migration considerations and scenarios that may not apply to all Commerce Suite 5.1 customers. In most cases, these sections should be considered as optional steps. They include:

- Chapter 7, "Member subsystem migration considerations," on page 133
- Chapter 8, "Access control subsystem considerations," on page 137
- Chapter 9, "Payment component migration considerations," on page 143
- Chapter 10, "Other migration considerations," on page 149

Chapter 7. Member subsystem migration considerations

This chapter describes the migration scenarios depending on how a user decides to integrate LDAP with the WebSphere Commerce 5.4 database, when migrating the Member subsystem from Commerce Suite 5.1 to WebSphere Commerce 5.4.

> **Important**
>
> You should always back up your database before migration or before switching from using the database to using a directory server as the Member repository. For information on backing up databases, see "Backing up your databases" on page 28.
>
> If you are using DB2, after migrating the database, you should run the following SQL statement and commands to increase the size of the `stmheap` parameter:
>
> ```
> db2 update db cfg for database_name using stmtheap 60000
> db2stop
> db2start
> ```
>
> where *database_name* is the database name of the migrated database. If you do not increase the `stmtheap` parameter to an adequate value, you may not be able to delete a newly created access group.

Overview of migration procedures

This section provides an overview of the migration procedures for the Member subsystem.

The conventions when reading the table below are:

DB->DB
> Migrate from a Commerce Suite 5.1 database to a WebSphere Commerce 5.4 database.

DS->DS
> Migrate from a Commerce Suite 5.1 directory server to a WebSphere Commerce 5.4 directory server.

DB->DS
> Migrate from a Commerce Suite 5.1 database to a WebSphere Commerce 5.4 database first, then switch to using a directory server.
>
> The migration from using a database to using a directory server can be done at any time after release to release migration. Hence, the proper way to read the DB->DS scenario is to first read the DB->DB column, then read the DB->DS column following the number sequence.

In Commerce Suite 5.1, the `wcsadmin` user is provided in the bootstrap data for the database. However, if you used a directory server in Commerce Suite 5.1, no corresponding bootstrap was provided for `wcsadmin` in the directory server. So it is unclear in Commerce Suite 5.1 who uses a directory server where the entry for `wcsadmin` is located in the directory server. The migration procedure below does assume such an entry exists in the directory server. However, regardless of where

the `wcsadmin` user actually resides in the directory server, in the `MBRREL` table which captures the membership hierarchy, the parent member of `wcsadmin` within WebSphere Commerce 5.4 will be set to the `Root Organization`.

The tasks done by automated migration below is only a subset of all the tasks done for the Member subsystem. There are other tasks (mainly the migration of roles and member groups) which are not listed below. More information on the **DB->DS** migration can be found in the WebSphere Commerce 5.4 online help.

Table 9. Overview of member subsystem migration procedures

Migration Procedures	DB->DB	DS->DS	DB->DS	Remarks
Create records in the BUSPROF table for business users, filling in ORG_ID and ORGUNIT_ID properly. Fix PROFILETYPE for users in the USERS table. Refer to "Member subsystem" on page 11	manual	manual	See Remarks	Already done during DB->DB migration
Run the automated migration script:				
Add the STATE column to the MEMBER table and populate it.	auto	auto	See Remarks	(*) Everything done by the automated script for DB ->DS has been completed during DB->DB migration.
Add Root Organization to the database.	auto	auto	see (*)	
Change ORG_ID for the wcsadmin user in the BUSPROF table from -2000 to -2001. Change PROFILETYPE for wcsadmin in the USERS table from C to B	auto	auto	see (*)	
Set MEMBER_ID in the ORGENTITY table for those that were null before to -2001 (Root Organization).	auto	auto	see (*)	
Create and populate the MBRREL table.	auto	auto	see (*)	
Examine DNs for organizational entities generated by automated data migration. Change the DNs (distinguished name) in the script if necessary.	n/a	manual	manual	
Run script to populate DNs for organizational entities in the ORGENTITY table and users in the USERS table	manual	manual	manual	
Create all necessary suffixes in the directory server. These are the suffixes that will be needed when organizational entities are created automatically by Commerce Suite 5.1 in the directory server.	n/a	manual	manual	
Create the ldapentry.xml file. For DS->DS migration, ldapentry.xml should be based on the content of the ldapmap.xml file in Commerce Suite 5.1.	n/a	manual	manual	See the note below.

Table 9. Overview of member subsystem migration procedures (continued)

Migration Procedures	DB->DB	DS->DS	DB->DS	Remarks
For DB->DS migration, using Configuration Manager, switch to using the directory server. Manually edit the *instance_name*.xml file to set the MigrateUsersFromWCSdb option to ON. By default, it is set to OFF.	n/a	n/a	manual	

Note: For DS->DS migration, in creating the ldapentry.xml, you have to do something slightly different in terms of specifying the user search bases depending on how you want your users to log on. The description below differentiates between two types of users:

- *DS users* refer to those users who exist in your directory server and are users that you want WebSphere Commerce to recognize. However, these users have never logged on to Commerce Suite 5.1 and have never been referenced in Commerce Suite 5.1. Therefore, they do not have entries in the WebSphere Commerce Suite database yet.

- *WCS DS users* refer to those users who exist in the your directory server and they are already recognized by Commerce Suite because they already logged on to the Commerce Suite 5.1 site. Such users have entries in the Commerce Suite database.

If you want both the DS users and WCS DS users to log on using RDN (relative distinguished name), both types of users must have RDN values such that they are all unique in the directory server. You then specify search bases such that both types of users can be found. During a search of the directory server for a user, WebSphere Commerce expects only one user to be found. If multiple users are found, it is an error situation.

If you want your DS users and WCS DS users to have the same RDN (for example, one DS user may have DN 'uid=john,o=IBM,c=US', another WCS DS user may have DN 'uid=john,o=CompanyA,o=Root Organization; notice that both users have RDN value 'john'):

- You can have your WCS DS users continue to log on with the same logon ID they used in WebSphere Commerce Suite 5.1. Have your DS users log on using a DN. You need to specify the search bases for your WCS DS users such that the search bases do not overlap with where your DS users reside.

Refer to the LDAP integration section of the WebSphere Commerce 5.4 online help for more information on DB->DS migration.

Existing Commerce Suite 5.1 user with existing directory server

In this scenario, you already have an existing directory server with some entries. You are using Commerce Suite 5.1 but not with your existing directory server. After migrating to WebSphere Commerce 5.4, you wish to use your existing directory server with WebSphere Commerce 5.4. Since you are a Commerce Suite 5.1 user, you have code running that uses only access beans to retrieve MEMBER data from the WebSphere Commerce 5.4 database.

For more information, see the section on "LDAP scenario: a database as the member repository" in the WebSphere Commerce 5.4 online help.

Continue using 5.1 directory server in WebSphere Commerce 5.4

In this scenario, you are already using a directory server with Commerce Suite 5.1. Now you migrate to WebSphere Commerce 5.4 and continue to use the directory server. As a Commerce Suite 5.1 customer, you have code running which uses only access beans to retrieve member data. Since you have used a directory server in 5.1, the registered users recognized by Commerce Suite 5.1 exist in the directory server and the users' data is replicated to the Commerce Suite 5.1 database, but organizational entities and member groups data reside only in the Commerce Suite 5.1 database.

In this scenario you need to do the following:

1. To ensure business users (B2B users) are properly placed within the membership hierarchy in WebSphere Commerce 5.4, you should ensure records exist in the BUSPROF table where necessary, with ORG_ID and ORGUNIT_ID set properly. In addition, set the profileType for users appropriately. You need to manually do the following:
   - If a business user's parent and ancestor organizational entities already exist in the Commerce Suite 5.1 database:
     – If the business user does not have a BUSPROF record, create a BUSPROF record for the business user and set ORG_ID and ORGUNIT_ID to point to the appropriate organizational entities.
     – If the business user already has a BUSPROF record, ensure the ORG_ID and ORGUNIT_ID are set properly in the BUSPROF record.
   - If a business user's parent and ancestor organizational entities do not exist in the Commerce Suite 5.1 database and you cannot create those organizational entities, consider setting the business user's profileType to C (B2C user).
2. Run the data migration script in "Migrating your database schema" on page 81 to:
   - Add the STATE column to the MEMBER table.
   - Change the ProfileType for the wcsadmin user from C to B.
   - Change ORG_ID in the BUSPROF table for wcsadmin from -2000 to -2001.
   - Add Root Organization to the database.
   - Fill in MEMBER_ID in the ORGENTITY table for those member Ids that were null in Commerce Suite 5.1.
   - Create the MBRREL table and populate it.
3. As part of automated data migration, a script is provided which will populate the ORGENTITY table with distinguished name (DN) values as described in "Updating distinguished names" on page 87. You should examine the DN values of the organizational entities to ensure they are what you want, and change any DN values as necessary. Then run the fillorgDN.sql script to populate the organizational entity DN values and also to populate the USERS table with DN values for registered users. Note that DN values are populated for registered users only and the DN values for the users will be replaced by WebSphere Commerce logic later on.
4. Ensure that all suffixes that are required are properly created in the Directory server. These suffixes are needed for when WebSphere Commerce automatically creates organizational entities in the Directory server.
5. Create an ldapentry.xml file based on ldapmap.xml, and add the mapping for organizational entity attributes into ldapentry.xml.
6. Ensure that the MigrateUsersFromWCSdb option in the instancename.xml file is OFF.

Chapter 8. Access control subsystem considerations

The access control model for WebSphere Commerce 5.4 is based upon the enforcement of access control policies. Access control policies are enforced by the access control Policy Manager. In general, when a user attempts to access a protectable resource, the access control policy manager determines if the user is allowed to perform the requested operation on the given resource.

In addition, note the following:

- The following Commerce Suite 5.1 access control database tables are deprecated in WebSphere Commerce 5.4:
  - ACCMBRGRP
  - ACCCMDGRP
  - ACCCUSTEXC
  - ACCCMDTYPE

  These tables are no longer used by the server runtime to determine access control. They have been replaced with several new access control tables. See ″Access Control″ in the WebSphere Commerce 5.4 online help for more details.
- Any changes made to the default Commerce Suite 5.1 bootstrap access control policies (ACCCMDGRP table) will be lost. However, any additions to this table are preserved and migrated appropriately to WebSphere Commerce 5.4 by the data migration process.
- In Commerce Suite 5.1, ACCCUSTEXC is an exclusion table that prevented specified Customer Commands from being executed within specified stores. Since WebSphere Commerce 5.4 follows a more permissive access control model, exercise due care to ensure that there are no *relaxed* policies higher in the member hierarchy.
- In WebSphere Commerce 5.4 the two levels of access control are:
  - Command level (coarse-grain).
    Determines whether a user has access to a Controller command or view.
  - Resource level (fine-grain) — this is also known as role-based access control.
    Determines whether a user can perform an action on a specific instance of a resource.

The following table describes the differences between Commerce Suite 5.1 access control, and WebSphere Commerce 5.4 access control. The main difference is that Commerce Suite 5.1 uses programmatic resource level access control, while WebSphere Commerce 5.4 uses policy-based, resource level access control. In order to minimize the failure of any customized code, the WebSphere Commerce runtime currently handles both the Commerce Suite 5.1 and WebSphere Commerce 5.4 access control related command methods. However, it is highly recommended that you migrate any customized code to use the WebSphere Commerce 5.4 methods, to make use of the policy-based access control model. Any given command should function properly if it is entirely using one of following the access control models:

- The recommended WebSphere Commerce 5.4 model using the validateParameters and getResources methods.
- The WebSphere Commerce Suite 5.1 model using the checkParameters, checkPermision, and getResourceOwners methods.

Table 10. Access Control Subsystem Differences

Item	Commerce Suite 5.1	WebSphere Commerce 5.4
Access Control Model	Role-Based Access Control In Commerce Suite 5.1, command level access control is implemented using the `ACCCMDGRP` table. Resource level access control is done programmatically in your source code. Changes in resource level policies required that you recompile your source code.	Policy-Based Access Control In WebSphere Commerce 5.4, command level and resource level access control is implemented using the `ACPOLICY` table. You can change policies without recompiling your source code.
Databeans	Protected programmatically	Protected directly and indirectly using the Delegator interface. If the databean does not implement this interface, it can be populated by anyone. Furthermore, even when the databean implements the Delegator interface, if it returns `null` in the `getDelgate` method, it can also be populated by anyone.
`getResources()`	N/A	This command method is used to trigger resource level access control checking. It returns all the protectable primary resources accessed by this command. It returns null if no resources are being accessed by this command.
`getResourceOwners()`	Default behavior: Returns the owner of the store if a valid store ID parameter is defined for the command, that is, from `requestProperties` or session. Returns `EC_ACC_ALL_RESOURCES` if no store ID is available, or if `storeId` is set to `ECConstants.EC_NO_STOREID`	The `getResourceOwners()` method returns `null` by default. In order to simulate Commerce Suite 5.1 behavior, when performing command level access control check, (that is, performing an access check where the command is the protectable resource) the command framework will use the resource owners returned from `getResourceOwners()`. It will also use the resource owners as the owner of the command. For examples, see "Examples using getResourceOwners()" on page 140.
`checkPermission()`	This method provides fine grain access control check. The WebSphere Commerce Suite Command framework invokes this method before the `performExecute()` method. Returns true if the administration command does not have fine grain access control.	The `checkPermission()` method has been kept for backward compatibility but should no longer be used for access control.

Table 10. Access Control Subsystem Differences (continued)

Item	Commerce Suite 5.1	WebSphere Commerce 5.4
checkParameters()	This is where Commerce Suite 5.1 performs parameter checking. The default implementation does not result in any action. The performExecute() of the ControllerCommandImpl and TaskCommandImpl calls checkParameters(). Most commands call super.performExecute() as the first line in their performExecute() in order to invoke checkParameters().	This method is replaced by validateParameters() in WebSphere Commerce 5.4 to support the new access control model. The default implementation does not result in any action. For backward compatibility, the performExecute() of the ControllerCommandImpl and TaskCommandImpl calls checkParameters(). Most commands call super.performExecute() as the first line in their performExecute() for good programming convention. This method, checkParameters(), will be deprecated in the next release.

Targetable commands	The sequence of method calls is:
	
	`{`
	`Command.checkPermission();...`
	`  Command.performExecute();...`
	`  Command.checkParameters();`
	`}`
		If you want to migrate your commands to the WebSphere Commerce 5.4 access control model, you need to implement validateParameters(). If you used checkParameters() in Commerce Suite 5.1, move your logic to validateParameters() and remove checkParameters() from your code.
		
		The sequence of method calls is:
		
		`{`
		`Command.validateParameters();`
		
		`Command.getResources();`
		
		`Command.checkPermission();`
		`// for backward compatiblity only`
		
		`Command.performExecute();`
		
		`Command.checkParameters();`
		`// for backward compatiblity only`
		`}`

Controller commands and views	In Commerce Suite 5.1, any controller commands that are defined in the URLREG table, but do not have a corresponding entry in the ACCCMDGRP table, are not under access control. As such, they are accessible by all users, including guest shoppers. Similarly, views that are defined in the VIEWREG table, but do not have a corresponding entry in the ACCCMDGRP table, are also accessible by all users. **Note:** A controller command or view that is in the ACCCMDGRP table, and has MbrGrp_Id = -2 , (that is, assigned to Customer Access Group), is also accessible by all users.	In WebSphere Commerce 5.4, the access control model has changed. Now, if a controller command does not explicitly have an access control policy that grants all users access to the command, ordinary users cannot access the command — only site administrators. Similarly, an explicit access control policy that grants access to a view is required if a user accesses the view directly from a URL or a command redirects to the view.

Notes:

1. If you have added a controller command that extends from a Commerce Suite 5.1 command, WebSphere Commerce 5.4 will only add the command level policy for it during migration. If the Commerce Suite command implements

getResources(), then you either have to determine what resources it is returning and create the appropriate resource level policy for your command, or, if you do not want resource level access control, you have to override getResources() on your command so that it returns a null value.

To determine what the WebSphere Commerce 5.4 commands are returning for their getResources(), analyze the trace and look for Action=*WCBECommand* and find all the Protectable resources getResources() is checking for. In the trace above, the resource is Order.For example, consider if after enabling the SERVER trace, you find the following in the logs:

```
============== TimeStamp:    2001-11-16 02:42:30.937
Thread ID:    <Worker#3>
Component:    SERVER
Class:        AccManager
Method:       isAllowed
Trace:        isAllowed? User=10012; Action=com.fvt.ACCOrderItemAddCmd;
Protectable=com.ibm.commerce.order.objects._Order_Stub;
  Owner=7000000000000002000resource is Groupable

==============
TimeStamp:    2001-11-16 02:42:30.984
Thread ID:    <Worker#3> Component:
SERVER Class: AccManager
Method:       isAllowed
Trace:        PASSED? =false
```

The meaning of the above trace is that the resource level policy is failing. In this case, ACCOrderItemAddCmd is extending from the Server OrderItemAdd command which implements getResources(). Therefore, by default, ACCOrderItemAdd also requires a resource level policy, unless the getResources() on it is changed to return null. This resource level policy is not added during migration, since it is not known what WebSphere Commerce 5.4 commands that you are extending.

In most cases, commands return access beans in the getResources() method. For example, returning com.ibm.commerce.xyz.objects.XYZAccessBean in getResources() will appear as com.ibm.commerce.xyz.objects._XYZ_Stub in the trace. This difference is because WebSphere Commerce 5.4 must narrow the access bean to its remote interface (since it is the remote interface of the EJB that actually extends the Protectable interface).

2. In WebSphere Commerce Suite 5.1, resource level access control was enforced programmatically, within the command logic. In WebSphere Commerce 5.4, resource level access control policies are specified externally, similar to the way command level access control policies are specified. During the migration, only command level access control policies are migrated from Commerce Suite 5.1 to WebSphere Commerce 5.4. Any resource-level access control policies that are needed due to customization of the Commerce Suite 5.1 default access control policies (which are stored in the ACCCMDGRP table), need to be added manually. Otherwise you will receive an unexpected access control violation exception. For details, see the related item in Appendix H, "Troubleshooting," on page 211.

Examples using getResourceOwners()

WebSphere Commerce Suite 5.1 commands:

- Commands that rely on default behavior.

  These commands do not implement getResourceOwners(), the default returns null.

When performing command level access control checks, the command framework returns the store owner as the command owner. It will return the `EC_SITE_ORGANIZATION` when no store ID is available.

- Commands that implement the `getResourceOwners()` method

  When performing the command level access check, the command framework performs an access check on the command for each resource owner returned by `getResourceOwners()`. For example, suppose that the `getResourceOwners()` method returns 2 owners: Organization 1 and Organization 2. The command framework will perform an access check on the command, first with Organization 1 as the owner. If this check passes, it will perform another access check on the same command, this time it will use Organization 2 as the command owner. It must pass both access checks.

New WebSphere Commerce 5.4 commands:

- These commands do not implement `getResourceOwners()`; the default returns `null`.
- When performing a command level access control check, the command framework will return the store owner as the command owner. It will return the `EC_SITE_ORGANIZATION` when no store ID is available.

Chapter 9. Payment component migration considerations

This section documents migration considerations and actions for the Payment component.

Migrating from PAYMTHD table to payment policies

Commerce Suite 5.1 uses three database tables to define payment methods supported by a store or store group. These tables are:

PAYMTHD
> The Payment Method table is a site-wide table that lists all the cashier profiles used in the mall. Each profile has a unique integer ID and a name.

PAYMTHDDSC
> The Payment Method Description table is a site-wide table that contains a short description of each of the Commerce Suite 5.1 cashier profiles in each of the supported language

PAYMTHDSUP
> The Payment Method Supported table lists all the profiles support by the store or store Group.

Commerce Suite 5.1 shipped with five entries in the PAYMTHD table and five cashier profiles. The five entries in the PAYMTHD table are summarized in the following table.

PAYMTHD_ID	PROFILENAME	Remark
100	WCS51_CustomOffline	Standard Commerce Suite 5.1 profile for the CustomOffline cassette.
200	WCS51_OfflineCard	Standard Commerce Suite 5.1 profile for the OfflineCard cassette.
300	WCS51_SET_MIA	Standard Commerce Suite 5.1 profile for the Cassette for SET (Secure Electronic Transactions) using the MIA (Merchant Initiated Authorization) SET extension.
400®	WCS51_SET_Wallet	Standard Commerce Suite 5.1 profile for the Cassette for SET using Wallet.
500	WCS51_CyberCash	Standard Commerce Suite 5.1 profile for the Cassette for CyberCash.

These payment methods are limited to payment methods supported by WebSphere Payment Manager.

Business policies and business policy commands

WebSphere Commerce 5.4 introduces the concept of *business policies* and *business policy commands*.

One category of business policies is the payment business policy (or in short, payment policy). Payment policy defines a set of business policy command interfaces that WebSphere Commerce 5.4 calls to perform business functions related to the business policy. Different payment policies may have different business policy command implementation.

WebSphere Commerce 5.4 payment policies are not limited to payment methods supported by the WebSphere Payment Manager and thus are more general than the payment methods defined for Commerce Suite 5.1.

When migrating from Commerce Suite 5.1 to WebSphere Commerce 5.4, you have to change from specifying `payMethodId` on the `OrderProcess` command to specifying `policyId` if you wish to use new WebSphere Commerce 5.4 functionality or payment behavior. Use the following table to help you make the appropriate change:

Commerce Suite 5.1		WebSphere Commerce 5.4		
Pay method ID	Profile name	Policy ID	Policy name	Profile name
100	WCS51_CustomOffline	$xxxx^1$	CustomOffline_COD	WC51_CustomOffline_COD
		$xxxy^1$	CustomOffline_BillMe	WC51_CustomOffline_BillMe
200	WCS51_OfflineCard	200	OfflineCard	WC51_OfflineCard
300	WCS51_SET_MIA	300	SET_MIA	WC51_SET_MIA
		301	SET_MIA_PCard	WC51_SET_MIA_PCard
400	WCS51_SET_Wallet	400	SET_Wallet	WC51_SET_Wallet
500	WCS51_CyberCash	500	CyberCash	WC51_CyberCash
		600	VisaNet	WC51_VisaNet
		601	VisaNet_PCard	WC51_VisaNet_PCard
		700	BankServACH	WC51_BankServACH

Note: [1] Policies for the WCS51_CustomOffline profile name are not supplied. You need to create your own. See the section "Adding a new payment business policy" in the WebSphere Commerce 5.4 online help.

Note: Note that the `policyId` of the predefined payment policies is the same as the `payMethodId` of PayMethods used in Commerce Suite 5.1, so you can use the same value. You can choose to implement either the Commerce Suite 5.1 or WebSphere Commerce 5.4 payment behavior by using the appropriate parameters when calling the `OrderProcess` command:

- Using the `payMethodId` parameter invokes the `DoPaymentMPFCmdImpl` class to conform to the Commerce Suite 5.1 behavior using the `PAYMTHD`, `PAYMTHDDSC`, and `PAYMTHDSUP` tables. This assumes that you will not use the new functions or commands such as ATP inventory, Release To Fulfillment, Balance® Payment, and so on that are available in WebSphere Commerce 5.4. To use the new functions or commands, you must switch from using the `payMethodId` parameter to use the `policyId` parameter.

- Using the `policyId` parameter invokes the `DoPaymentMPFCmdImpl` class to conform to WebSphere Commerce 5.4 behavior using the `POLICY` and `POLICYCMD` tables.

  For example, in the migrated InFashion store, if you use ATP, replace the following line in `OrderDisplayPending.jsp`:

  ```
  <input type=hidden name="<%= ECConstants.EC_PAYMTHDID %>" value="200">
  ```

with

```
<input type=hidden name="policyId" value="200">
```

Otherwise, some functions will not work such as PickPatches in Commerce Accelerator.

In addition, if you are using the `ProfileCassetteAccountDataBean` data bean in Commerce Suite 5.1, you should switch to use the `UsablePaymentTCListDataBean` data bean in WebSphere Commerce 5.4. The `ProfileCassetteAccountDataBean` data bean uses the Commerce Suite 5.1 tables, `PAYMTHD`, `PAYMTHDDSC`, and `PAYMTHDSUP`. The `UsablePaymentTCListDataBean` data bean uses the new WebSphere Commerce 5.4 tables, `POLICY` and `POLICYCMD`.

Processing an order using the `OrderProcessCmd` controller command invokes the `DoPaymentMPFCmdImpl` class, which is the standard implementation class for the `DoPaymentCmd` task command for WebSphere Commerce 5.4. For detailed information on how payments are processed in WebSphere Commerce 5.4, see the sections on WebSphere Payment Manager in the online help. For detailed information on the above interfaces, see the online help.

▶ Business If you are migrating to a B2B store that uses contracts with specified payment terms and conditions, you can use the WebSphere Commerce 5.4 Commerce Accelerator to create the account, contracts, and the payment terms and conditions. For such cases, the `tcId` parameter is also required to identify the Payment terms and conditions to be used. You can use the `UsablePayementTCListDataBean` data bean to return the `tcId` associated with the Payment terms and conditions.

See the WebSphere Commerce 5.4 online help for additional information on the payment policies. WebSphere Commerce 5.4 includes several other payment policies in addition to the ones listed here.

Migrating to WebSphere Commerce 5.4 business policy commands for payment

WebSphere Commerce 5.4 specifies a set of business policy command interfaces for the payment policy:

- `DoPaymentPolicyCmd`
- `CheckPaymentAcceptPolicyCmd`
- `DoDepositPolicyCmd`
- `DoRefundPolicyCmd`
- `DoCancelPolicyCmd`

Different Payment Policies may have different implementations for these commands.

WebSphere Commerce 5.4 defines two sets of implementation classes for the payment policies included with WebSphere Commerce 5.4. One set supports the payment policies that are based on the WebSphere Payment Manager, and the other set supports the credit line payment policy which is non-WebSphere Payment Manager based. The business policy command implementation classes for payment policies that are based on WebSphere Payment Manager are:

- DoPaymentPMCmdImpl
- CheckPaymentAcceptPMCmdImpl
- DoDepositPMCmdImpl
- DoRefundPMCmdImpl
- DoCancelPMCmdImpl

The business policy command implementation classes for the credit line payment policy are:
- DoPaymentCLCmdImpl
- CheckPaymentAcceptCLCmdImpl
- DoDepositCLCmdImpl
- DoRefundCLCmdImpl
- DoCancelCLCmdImpl

The appropriate business policy command implementation classes are called, depending on the payment policy used.

For more information on the above implementation classes, see the WebSphere Commerce 5.4 online help.

Changing CMDREG entry for DoCancelCmd

If you use the InFashion Store in Commerce Suite 5.1 or have created your own store that uses the WebSphere Payment Manager, you need to change the CMDREG entry for DoCancelCmd to make it work in WebSphere Commerce 5.4. In Commerce Suite 5.1, DoCancelCmd is assigned to the class com.ibm.commerce.payment.commands.DoCancelPMCmdImpl if WebSphere Payment Manager is used. With WebSphere Commerce 5.4, DoCancelCmd should be assigned to class com.ibm.commerce.payment.commands.DoCancelCmdImpl. The DoCancelCmdImpl interface routes the call to DoCancelPMCmdImpl or DoCancelCLCmdImpl depending on the payment policy used for payment.

Note that this change is done automatically when you run the data migration script, described in "Migrating your database schema" on page 81.

Sample JavaServer Pages file - PayStatusPM.jsp

The sample JavaServer Pages file PayStatusPM.jsp in the /usr/WebSphere/CommerceServer/samples/web/payment directory has been updated slightly for Payment Manager 3.1.2. The changes are required for the JSP to generate a correct status message related to the state of the payment of an order.

In the previous release, the PayStatusPM.jsp treats the WebSphere Payment Manager order state of Refundable as an indication that the payment for the order has progressed beyond the approved state. Therefore, it generates the status message assuring a shopper that their order "has been authorized.....". This was true with the Payment Manager cassettes shipped with Commerce Suite 5.1.

However, with Payment Manager 3.1.2, some of the Cassettes set the WebSphere Payment Manager order state to the Refundable state as soon as the WebSphere Payment Manager order has been created, and before any payment has been approved. This has necessitated the change in PayStatusPM.jsp to enable it to generate the correct status message.

The WebSphere Payment Manager Cassettes that exhibit this behavior are:
- Cassette for CyberCash
- Cassette for VisaNet
- CustomOffline Cassette
- OfflineCard Cassette

If you have your own version of the `PayStatusPM.jsp` and you plan to use the above Cassettes with WebSphere Commerce 5.4, you should make the corresponding change to your JSP so the correct status message is displayed to the shopper when they view the page. (Remember that the WebSphere Payment Manager order state of `Refundable` may not mean that the payment has already been *approved*.)

Refer to the `PayStatusPM.jsp` file that is supplied with WebSphere Commerce 5.4 for details. It is located in the `/usr/WebSphere/CommerceServer/samples/web/payment` directory.

Chapter 10. Other migration considerations

This chapter describes several other areas of WebSphere Commerce 5.4 of which you should be aware, and take action, if required.

Default currency behavior

To allow your customers to select their preferred currency in which to shop, you can include a list of currencies that you support for payment on store pages.

- If the customer's preferred currency is supported by the store, it then becomes the shopping currency.
- If the preferred currency is not supported and if there is another currency which has the customer's preferred currency as a valid counter value, then that currency is used as the shopping currency.
- If there is no other currency with the preferred currency as a counter value, then the store's default currency is determined from the new SETCURR column in the STOREENT table. This setting is independent of the customer's language ID. Note that this behavior is new for WebSphere Commerce 5.4.
- If there is no other default currency specified in the STOREENT table, then the store's default currency for the customer's language ID is used. In this case, the STORELANG database table determines the settings.

Notes:

1. Only shopper's with no preferred currency or a non-supported preferred currency (which is not the counter value for a supported currency) are affected by this change. If the shopper has a supported preferred currency, they will always see this currency no matter what language they choose.

2. To maintain the previous Commerce Suite 5.1 default currency in WebSphere Commerce 5.4, do not set the store's default currency in the STOREENT table. To implement the new WebSphere Commerce 5.4 default currency, set the store default in the STOREENT table. You do not need to change the STORELANG table parameters from Commerce Suite 5.1. Thus, you can choose which migrated stores implement the new default currency behavior, and which stores maintain the previous behavior.

3. It is recommended that you set the SETCURR column of the STORELANG table to null as it may be deprecated in the future.

4. Set the SETCCURR column of the store or the store group in the STOREENT table. For migrated stores, this will not be the case initially. Any new stores should set the store or store group default currency.

Considerations for pricing

The following are the new commands and methods in WebSphere Commerce 5.4 that replace the Commerce Suite 5.1 commands for pricing:

Task commands

- GetContractUnitPriceCmd replaces GetBaseUnitPriceCmd
- GetContractSpecialPriceCmd replaces GetBaseSpecialPriceCmd
- GetProductContractUnitPriceCmd replaces GetProductBaseUnitPriceCmd

Note: For backward compatibility, the Commerce Suite 5.1 commands are maintained in WebSphere Commerce 5.4.

Data beans

For the following data beans, in Commerce Suite 5.1, the method `getCalculatedPrice()` was available to retrieve the price. In WebSphere Commerce 5.4, it will be replaced with a new method, `getCalculatedContractPrice()`.

- ItemDataBean
- PackageDataBean
- ProductDataBean
- CatalogEntryDataBean
- InterestItemDataBean
- BundleDataBean

Note: For backward compatibility, the Commerce Suite 5.1 methods will be maintained in WebSphere Commerce 5.4.

For more information on the above commands or method, see the WebSphere Commerce 5.4 online help.

Part 3. Appendixes

Appendix A. Starting and stopping WebSphere Commerce Suite 5.1 components

This appendix explains how to start and stop each product that is provided as part of the WebSphere Commerce Suite package, as well as the optional Web server, Domino Web Server. Use this information if you need to restart any component.

Starting and stopping WebSphere Commerce Suite

To start or stop your WebSphere Commerce Suite instance, do the following:

1. Log in as user ID `root`.
2. Ensure that your database management system and WebSphere Application Server are started. If you are using DB2, refer to "Starting and stopping DB2 Universal Database" on page 154. For WebSphere Application Server, refer to "Starting and stopping WebSphere Application Server."
3. Launch the WebSphere Application Server Administrator's console by entering the following in a terminal window:

   ```
   cd /usr/IBMWebAS/bin
   ./adminclient.sh
   ```

4. Expand **WebSphere Administrative Domain**.
5. Expand your host name.
6. Select **WebSphere Commerce Server —***instance_name* and right-click. Select **Start** or **Stop** as appropriate.

Starting and stopping WebSphere Application Server

To start WebSphere Application Server, do the following while logged in as user ID `root`:

1. Ensure that your database management system is started.
2. Type the following commands in a terminal window:

   ```
   cd /usr/IBMWebAS/bin
   ./startupServer.sh &
   ```

   Check the `/usr/IBMWebAS/logs/tracefile` to ensure that WebSphere Application Server started successfully.

To stop WebSphere Application Server, do the following while logged in as user ID `root`:

1. Start the WebSphere Application Server administration client by entering the following in a terminal window:

   ```
   cd /usr/IBMWebAS/bin
   ./adminclient.sh
   ```

2. In the WebSphere Application Server Administrative Console, select the node with the name of your unqualified host name.
3. Click the stop button. You will get a warning message similar to the following:

   You are trying to stop the node that the console is connected to. This will cause the console to exit after the node is stopped. Do you want to continue?

   Click **Yes** to continue.

4. After the WebSphere Application Server Administrative Console, confirm that all WebSphere Application Server–related processes have stopped by issuing the following command in a terminal window:

   ```
   ps -ef | grep IBMWebAS
   ```

5. If this command returns any Java processes, stop them by issuing a kill command for each of processes found.

Starting and stopping IBM HTTP Server

To start IBM HTTP Server, while logged on as user ID root, enter the following commands in a terminal window:

```
cd /usr/HTTPServer/bin
./apachectl start
```

To stop IBM HTTP Server, while logged on as user ID root, do the following:

1. Ensure that WebSphere Commerce Suite and WebSphere Application Server are stopped.

2. Enter the following commands in a terminal window:

   ```
   cd /usr/HTTPServer/bin
   ./apachectl stop
   ```

> **Important**
>
> A core dump will occur if you attempt to stop the IBM HTTP Server before stopping WebSphere Commerce Suite and WebSphere Application Server.

To start IBM HTTP Administration Server, while logged on as user ID root, enter the following commands in a terminal window:

```
cd /usr/HTTPServer/bin
./adminctl start
```

To stop IBM HTTP Administration Server, while logged on as user ID root, enter the following commands in a terminal window:

```
cd /usr/HTTPServer/bin
./adminctl stop
```

Starting and stopping DB2 Universal Database

To start DB2, do the following:

1. Log on as your DB2 instance ID.

2. Type db2start.

To stop DB2, do the following:

1. Stop Commerce Suite according to the instructions in "Starting and stopping WebSphere Commerce Suite" on page 153.

2. While logged on as your DB2 instance ID, type db2stop. If any applications are connected to DB2, use the following command instead:

   ```
   db2stop force
   ```

Starting and stopping WebSphere Payment Manager

To start WebSphere Payment Manager, ensure your database started, then start the payment engine as described in "Start the WebSphere Payment Manager engine," and then your Web server. Ensure that IBM WebSphere Application Server 3.5.3 is running.

Start the WebSphere Payment Manager engine

When you start the payment engine using the `IBMPayServer` script, you will need to specify the database administrator password. The database administrator password can either be specified as an argument on the command line or, more securely, using a `.payment` file in the directory where WebSphere Payment Manager is installed. If you specify the password on the command line, the password used to start the payment engine will be visible to anyone with access to the machine where the WebSphere Payment Manager is installed. If you do not enter the database administrator password on the command line, the WebSphere Payment Manager tries to read the `.payment` file to obtain the password.

Note: When using DB2 Universal Database as your WebSphere Payment Manager database, remember that DB2 Universal Database always uses your operating system user ID and password as its database administrator user ID and password. When prompted to supply your database administrator password at payment engine startup (either in the `.payment` file or at the command line), you should enter your operating system database administrator password.

Start the payment engine with a .payment file

Using the `.payment` file for password retrieval protects the password, and thus your payment data, from anyone with access to the WebSphere Payment Manager machine. The WebSphere Payment Manager installation program creates the `.payment` file in the directory where WebSphere Payment Manager is installed (except when migrating from prior releases). The `.payment` file is created as read-only and is hidden as much as the operating system allows.

To start the payment engine with a `.payment` file:

1. Make sure the `.payment` file has the correct database password.
2. Enter the following commands from a terminal window:

   ```
   cd /opt/PaymentManager
   ./IBMPayServer
   ```

Note: The payment engine will stop if you log off after completing these steps. To keep the payment engine running in the background after you log off, use the **nohup** command as follows: `nohup ./IBMPayServer &`. When using the **nohup** command, any messages written to `STDOUT` by the payment engine will be placed in a nohup.out file. This file will be in the directory where you ran the **nohup** command.

Start the WebSphere Payment Manager user interface

After starting the WebSphere Payment Manager engine and WebSphere Application Server, to start the WebSphere Payment Manager user interface:

1. Point your web browser to:

   ```
   http://<host_name>/PaymentManager/
   ```

where *<host_name>* is the machine where you installed WebSphere Payment Manager.

2. On the WebSphere Payment Manager Logon window, enter the WebSphere Payment Manager administrator's user ID and password, and click **OK**. The default user ID and password are both `wcsadmin`. For information on creating WebSphere Payment Manager user IDs, see the latest *IBM WebSphere Payment Manager for MultiPlatforms, Administrator's Guide*.

You can also access a subset of the WebSphere Payment Manager user interface function from the WebSphere Commerce Suite Administration Console.

Stop the WebSphere Payment Manager

To stop the WebSphere Payment Manager, you need to:
- Stop the payment engine
- Stop the WebSphere Payment Manager application server under WebSphere Application Server
- Stop all other WebSphere Payment Manager servlets.

Stopping the payment engine

You can stop the payment engine using the `StopIBMPayServer` command. There are no arguments to the `StopIBMPayServer` script. You must have a `.payment` file in the WebSphere Payment Manager install directory to use this command. The `StopIBMPayServer` command is found in `/opt/PaymentManager`.

To stop the payment engine without using the `StopIBMPayServer` command:
1. Find the process ID for the payment engine using the **ps** command.
2. Use the **kill** command to stop that process ID.

Stop WebSphere Payment Manager servlets using WebSphere Application Server

When using WebSphere Application Server 3.5, you can stop all servlets by stopping the WebSphere Payment Manager application server. To stop the WebSphere Payment Manager application server:
1. Go to the WebSphere Application Server Administration Client.
2. Select the `WebSphere Payment Manager` application server.
3. Right-click on the application server and select **Stop**.
4. Exit the Administration Client.

Starting and stopping Domino Web Server

To start Domino HTTP Web Server, from the Lotus® Domino Server command prompt, type:
```
load http
```

To stop Domino HTTP Web Server, from the Lotus Domino Server command prompt, type:
```
tell http quit
```

Appendix B. Extending the data migration script

If you have extended the standard WebSphere Commerce Suite 5.1 database schema, you will need to update the supplied data migration scripts as follows. For example, consider the case if you have added a new column called `FRIEND` to the standard `USERREG` table, as `INTEGER DEFAULT 0` and `FRIEND` has a foreign key to the `MBRGRP` table.

Before starting the automatic data migration using the supplied migration scripts:

1. **For DB2 databases**, proceed to the `/usr/WebSphere/CommerceServer/schema/db2` directory.

   For Oracle databases, proceed to the `/usr/WebSphere/CommerceServer/schema/oracle` directory.

2. Make a back up copy of the file `wcs.schema.sql`.

3. Edit the `wcs.schema.sql` file and look for the `CREATE TABLE USERREG` statement:

   ```
   CREATE TABLE USERREG (
                     ...
                        );
   ```

4. Add the new column `FRIEND` to the `CREATE TABLE USERREG` statement as follows:

   ```
   CREATE TABLE USERREG  (
                     ...
      FRIEND INTEGER DEFAULT 0
                           );
   ```

   Ensure that the format of the file is maintained. The database migration script will parse the file and assume that each column definition within the table is an unique line and there are no blank lines among them. (This includes the brackets.)

5. Make a back up copy of the file `wcs.referential.sql` in the same directory.

6. Edit the file `wcs.referential.sql` and add the referential constraint to the file:

   ```
   ALTER TABLE USERREG ADD
     CONSTRAINT F_FRIEND FOREIGN KEY (FRIEND) REFERENCES MBRGRP ON DELETE CASCADE
   ```

7. Check your tables to see if there are other constraints from any new tables that you have created that are pointing to the standard WebSphere Commerce schema tables. If there are, you should drop these constraints before running the data migration scripts.

   Note: You need to recreate these constraints yourself after completing the migration.

8. Check the `MapFile.txt` in the `/usr/WebSphere/CommerceServer/schema/migration/`*FromRelease* directory, where *FromRelease* represents the version you are migrating from:

   - 5.1.0.1 (for WebSphere Commerce Suite 5.1)
   - 20011015 (for WebSphere Commerce Business Edition 5.1)
   - 20011215 (for WebSphere Commerce 5.4 Limited Availability release)

   If the table you have modified is listed in `MapFile.txt`, you need to add the additional column to both the source and target column lists. For example, consider the entry for the `ADDRBOOK` table in `MapFile.txt`. (Note that the following example has been split across lines for display purposes):

```
addrbook=(addrbook_id, description, displayname, member_id, type);
         (ADDRBOOK_ID, DESCRIPTION, DISPLAYNAME, MEMBER_ID, TYPE)
```

The first set of character strings enclosed in parentheses represent the columns
for the ADDRBOOK table in the Commerce Suite 5.1 database. The second set of
character strings enclosed in parentheses represent the columns for the
ADDRBOOK table in the WebSphere Commerce 5.4 database. Suppose that you
have added the FRIEND column to the ADDRBOOK table in Commerce Suite 5.1. In
that case, you also need to modify MapFile.txt to address the FRIEND column
as shown below in bold:

```
addrbook=(addrbook_id, description, displayname, member_id, type, friend);
         (ADDRBOOK_ID, DESCRIPTION, DISPLAYNAME, MEMBER_ID, TYPE, FRIEND)
```

Otherwise, the migration script will not migrate the FRIEND column over to
WebSphere Commerce 5.4.

9. Run the migration scripts as described in Chapter 4, "Migrating your database,"
   on page 69.

Appendix C. Overview of the migration script

This section describes what the migration script does and is included for your information. You are not required to take any action.

The database migration script makes a back up copy of your database into the directory that you specify, and then migrates the following subsystem or components:

- Member
- Catalog
- ATP inventory
- Order items
- Contracts
- Campaigns
- Access control

Member migration

The migration script performs the following updates on the Member subsystem:

- The script sets the state of the MEMBER table as follows:
  - The script sets the state to null (no state) for:
    - Guest users (user registration type set to G)
    - member groups
  - The script sets the state to Approved (1), for:
    - Registered users (user registration type set to R)
    - Site administrators (user registration type set to S)
    - Administrators (user registration type set to A)
    - Organizational entities

  The possible states of the MEMBER table are as follows:

  0   Pending Approval

  1   Approved

  2   Rejected

- Sets the OWNER_ID column of the MBRGRP table to -2001 (that is, the Root Organization) if it is not set (that is, contains a value of 0).
- Sets the users' profile type.
  - For users that are registered as user type S or A (site administrators or administrative role) in Commerce Suite 5.1, the script sets the PROFILETYPE to B
  - For users that have a business profile (that is, the BUSPROF table is setup) and that have a null PROFILETYPE in Commerce Suite 5.1, the script sets the PROFILETYPE to B, as it assumes they are B2B users.

    For example, if the user's PROFILETYPE is set to C (a B2C user) the script does not reset the profile type.
- Checks the MBRGRP table.

If the `MBRGRPUSG` table has a `MBRGRPTYPE_ID` of `AccessGroup`, the script creates a corresponding role in the `ROLE` table with the following exceptions:

– If the `MBRGRPTYPE_ID` is set to -2 (`CustomerGroup`) since the Customer Group is not a role in WebSphere Commerce 5.4.

– If the Order Clerk role is used in Commerce Suite 5.1, the migration script migrates it to a user defined role in WebSphere Commerce 5.4.

The script moves all roles in the WebSphere Commerce 5.4 `ROLE` table to the `MBRROLE` table and sets `MEMBER_ID` to a value of -2001 (`Root Organization`). The Root Organization can access all these roles.

• Checks the `ORGENTITY` table and sets the parent `MEMBER_ID` to -2001 (`Root Organization`) if the `MEMBER_ID` is null.

• Checks all users with a register type of `S`. The script:

– Creates an entry in the `MBRROLE` table with the Site administrator role set to -1.

– Ensures that all the parents and ancestors have the same role assigned.

– If users have a register type of `A`, the script explicitly migrates them to an administrator group (which has some predefined roles associated with the administrator group).

– Create an entry in the `MBRGRPMBR` table with the `MBRGRP_ID` pointing to the new administrator group.

• For each record in the `ACCMBRGRP` table, the script:

– Adds a record to the `MBRROLE` table.

– Adds an additional record the `MBRROLE` table for the parent organizational entity to which the administrator belongs. If the `OWNER_ID` is 0, then the script sets it to `-2001`.

• Builds the `MBRREL` table, but only for registered users, and not for guest users where `MEMBER_ID` in the `ORGENTITY` table is set to 0 (zero).

• Generates the SQL statements to fill the `DN` of the `ORGENTITY` table.

The script generates a file called `fillorgDN.sql`. You can either manually update the DN (distinguished name) column of the `ORGENTITY` table, or use this file to update the table. See "Updating distinguished names" on page 87 for more information.

Catalog migration

The migration script performs the following updates on the Catalog subsystem:

• Creates a parent product for items that do not have any parent.

– It creates a `CATENTRY` table entry of the product type; the values for this entry are copied from the item `CATENTRY`.

– Creates a `CATENTDESC` table entry (the catalog entry description).

– Creates the `CATENTREL` table entry (the product-item relationship)

• Creates fulfillment entries for products and items.

– Creates a `BASEITEM`, `BASEITEMDSC`, `ITEMVERSN` and `STOREITEM` table entry for each product.

– Creates an `ITEMSPC` and `VERSIONSPC` table entry for each item.

• Creates fulfillment entries for packages.

– Creates a `BASEITEM`, `BASEITEMDSC`, `ITEMVERSN` and `STOREITEM` table entry for each package.

– Creates an `ITEMSPC` and `VERSIONSPC` table entry for each package.

ATP inventory migration

The migration script performs the following updates on the Inventory subsystem:

* If you choose to migrate to the new ATP inventory representation, the script updates `ALLOCATIONGOODFOR` to a value of `43200` in order to enable ATP support.

  If you choose not to migrate to the new ATP inventory representation, the script updates `ALLOCATIONGOODFOR` to a value of `0` to postpone ATP inventory migration and continue tracking inventory as Commerce Suite 5.1 does, using the `INVENTORY` table.

  Refer to the database schema documentation for the `STORE` table in the WebSphere Commerce 5.4 online help to understand the meaning of this value. Any nonzero value will cause ATP inventory to be enabled.

* Creates the entries for the `DISTARRANG` table for each product. It sets the `ENDDATE` to a large value (59 years) and the `STARTDATE` to the current date.

* Creates the entries for the `ITEMFFMCTR`, `RADETAIL`, `RECEIPT`, and `RCTAVAIL` tables for each item.

* Create the entries for the `ITEMFFMCTR`, `DISTARRANG`, `RADETAIL`, `RECEIPT`, `RCTAVAIL`, tables for each package.

Note that when you migrate to WebSphere Commerce 5.4 specifying `atp`, the database migration script does not populate any vendors in the `RECEIPT` table during the migration as there be multiple vendors per store or there may be none. If you do not have a vendor assigned to your store, you should create one using the WebSphere Commerce Accelerator before you migrate to the ATP inventory representation..

Order items migration

The migration script checks all the orders that have a status of `P`, `I`, or `M`.

* Looks for all the order items under those orders

* Fills in the `ITEMSPC` field corresponding to the `CATENTRY_ID` in the `ORDERITEMS` table.

* Fills in the `TRADING_ID` field of all order items to use the store's default contract.

* Generates a script to convert all orders and order items whose status is `C` to a status of `S`.

Shipping calculation codes

In Commerce Suite 5.1, shipping calculation codes were calculated separately for groups of order items with different shipping addresses. That is, shipping calculation codes grouped their order items by shipping address. Grouping by shipping address is now optional behavior. To ensure backward compatible behavior, the migration script sets the `perAddress` flag of the `CALCODE.GROUPBY` column for all shipping calculation codes (those where `CALCODE.CALUSAGE_ID = -2`).

Payment migration

For each `CMDREG` entry, the migration script changes `DoCancelPMCmdImpl` to `DoCancelCmdImpl`. See "Changing CMDREG entry for DoCancelCmd" on page 146 for additional information.

Discount data migration

The migration script determines if discount data exists that was created by the Commerce Accelerator tool in Commerce Suite 5.1. For the Discount subsystem:

- If the script detects that discount data was created by tools other than the Commerce Suite 5.1 Commerce Accelerator, it leaves the discount data is as-is. It does not migrate the discount data to the WebSphere Commerce 5.4 level. In this case, you need to use the same tools and methods you used in Commerce Suite 5.1 to access and display your discount data.

- If the script detects that discount data was created by the Commerce Accelerator tool, it migrates the discount data in the `CALCODE` and `CALCODEMGP` tables to that required by WebSphere Commerce 5.4.

- Deletes the previous discount data if it has been migrated. Non-migrated discount data remains as-is.

Contract migration

The migration script performs the following updates on the Contract subsystem:

- If you have not used store default contract in Commerce Suite 5.1, the script creates a store default contract as required for WebSphere Commerce 5.4.

  If you have used store default contract in Commerce Suite 5.1, the script migrates your store default contract to aWebSphere Commerce 5.4 store default contract. It creates the seller participant and a contract level participant if no member group pricing is available. For each `TRADEPOSCN` entry, if the `FLAGS` column is set to zero, it creates standard price contract terms and conditions. If the `FLAGS` column is non-zero, the script creates custom price contract terms and conditions.

- For each `MGPTRDPSCN` entry, it creates a trading position container level buyer participant. If `MBRGRP_ID` is zero, the buyer participant, `MEMBER_ID`, is set to null to indicate that all buyers are qualified.

- For each custom price contract terms and conditions, the script:
  - Creates a corresponding custom product set for custom price list.
  - Populates the data in the product set using the data from the price list.

- If there are multiple contracts, the script ensures that only one is activated as a default contract by setting `STOREDEF.CONTRACT_ID` to the appropriate value.

- Creates an entry in the `TRADING` table for each contract.

- Adds entries to the new `STORECNTR` table.

Default contract

WebSphere Commerce 5.4, has introduced terms and conditions to provide contract support. The migration process will create a default contract for your system which will have the same behavior and characteristics as the WebSphere Commerce Suite 5.1 business flows (for example, for shipping charges).

Normally, you do not need to take any actions during the migration process, since the default contract is automatically created. If you need to create additional contracts for your business processes, use the WebSphere Commerce Accelerator to do so after you have migrated to WebSphere Commerce 5.4. See the section "Creating a new contract" in the WebSphere Commerce 5.4 online help.

The database migration script creates the following items for a WebSphere Commerce 5.4 default contract:

- Policy (`POLICY` table) and policy description (`POLICYDESC` table) for the migrated store:
  - A standard price policy for the master catalog (per store) — Mastercatalog TC.
  - Shipping charged by seller policy (`POLICY_ID=-7001` bootstrap data)
  - Shipping charged by carrier policy (`POLICY_ID=-7002` bootstrap data)
  - Return charge policy -- fractional charge by days (create one per store)
  - Return approval policy -- approval by days (create one per store)

  In addition, the script creates four policy commands for return charge and return approval (per store)

  The migration script assumes that you will use the original payment policy (policy_id=-2001 bootstrap data), thus it does not create it.
- The terms and conditions created for the default contract for the store — one shipping TC (per contract)

  The returns and refund terms and conditions are not created in order that you can have your store up and running without having to create new JavaServer Pages.

  The details of the returns and refund are specific to each store and is new for WebSphere Commerce 5.4. If you need to deploy this feature, you will need to create a new contract terms and conditions for your store. See the section "Creating a new contract" in the WebSphere Commerce 5.4 online help.
- Contract participants:
  - Seller participant
  - One or more buyer participants (depending on `MGPTRDPSCN`)
  - One contract level buyer participant

Campaign migration

The migration script migrates campaign initiatives, e-Marketing Spots, and customer profiles to the WebSphere Commerce 5.4 format. The customer profiles that were stored in the `SEGMENT` table are moved to the `MBRGRP` table. Campaign initiative rules are mined out of the `BZRPENTSTG` table and stored in the `RULE` column of the `INITIATIVE` table. Each condition is stored as a separate initiative. e-Marketing Spots are moved from the `MPE` table to the `EMSPOT` table. The scheduling of the campaign initiatives is moved to the `INTVSCHED` table.

Access Control migration

The Access Control migration involves migrating the following components:
- Migrating the Commerce Suite 5.1 `ACCMBRGP` table to the WebSphere Commerce 5.4 `MBRROLE` table.
- Migrating the Commerce Suite 5.1 `ACCCMDGRP` table to the WebSphere Commerce 5.4 `ACPOLICY` table.
- Migrating the Commerce Suite 5.1 `ACCCUSTEXC` table to the WebSphere Commerce 5.4 `ACPOLICY` table.
- Migrating the Commerce Suite 5.1 Order Clerk access group as required.

The details are as follows:
1. In Commerce Suite 5.1, users are given access privileges by being assigned to various access groups (member groups with the member group type set to

AccessGoup) in the ACCMBRGRP table. In WebSphere Commerce 5.4, access control policies are also partially based on access groups. However, instead of being assigned to access groups directly, users are assigned to roles, in the MBRROLE table. Most of the bootstrap access groups implicitly refer to role assignments. For example, the Sellers access group includes everyone that has been assigned the Seller role through the MBRROLE table.

The following table maps the Commerce Suite 5.1 access groups to the WebSphere Commerce 5.4 roles and the WebSphere Commerce 5.4 access groups.

Table 11.

Commerce Suite 5.1 access group	WebSphere Commerce 5.4 role	WebSphere Commerce 5.4 access group
Site Administrator (-1)	Site Administrator (-1)	SiteAdministrators (-1)
Customer (-2)	Not needed as a role since AllUsers access group implicitly includes everyone.	AllUsers (-2)
Customer Service Representative (-3)	Customer Service Representative (-3)	CustomerServiceRepresentatives (-3)
Merchant (-4)	Seller (-4)	Sellers (-4)
Order Clerk (-5)	Not used in bootstrap	Not used in bootstrap
Store Administrator (-6)	Store Administrator (-6)	StoreAdministrators (-6)
Store Developer (-7)	Store Developer (-7)	StoreDevelopers (-7)
Merchandising Manager (-8)	Product Manager (-8)	ProductManagers (-8)
Marketing Manager (-9)	Marketing Manager (-9)	MarketingManagers (-9)

Note: Generally, the names of roles are singular and the names of access groups are plural.

Entries in the ACCMBRGRP table are migrated to the MBRROLE table, using the above mapping of Commerce Suite 5.1 access group IDs to WebSphere Commerce 5.4 role IDs. If you have created an access group on your Commerce Suite 5.1 system, the data migration script creates a corresponding role with the same name as the access group.

2. The data migration script migrates any entries that you have added to the Commerce Suite 5.1 ACCCMDGRP table appropriately to the WebSphere Commerce 5.4 ACPOLICY table. If the entry referred to a bootstrap access group, the script updates an existing WebSphere Commerce 5.4 bootstrap access control policy to include the command or view. If the entry refers to an access group that you have created in Commerce Suite 5.1, the migration script creates a new policy, along with the appropriate components: Action, ActionDescription, ActionGroup, ResourceCategory, ResourceGroup, and so on.

3. Migrating the Commerce Suite 5.1 ACCCUSTEXC table involves overriding the appropriate AllUsers template access control policy:

   - AllUsersExecuteAllUserCmdResourceGroup_TemplatePolicy for excluding controller commands for the specific organizational entity.
   - AllUsersExecuteAllUsersViews_TemplatePolicy for excluding views for the specific organizational entity.

   Then, the script creates an appropriate AllUsers policy for this organizational entity, depending on which commands and views are not to be accessible in stores owned by this organizational entity.

4. The Order Clerk role is not used in WebSphere Commerce 5.4. As such, WebSphere Commerce 5.4 bootstrap access control policies do not make reference to this role. However, if the customer has added some customized controller commands or views that are to be accessed by this role, then the script creates appropriate access control policies in the ACPOLICY table.

Appendix D. Converting to ATP inventory at a later time

When migrating to WebSphere Commerce 5.4 using the database migration script, your inventory data is converted by default to the new ATP inventory representation supported by WebSphere Commerce 5.4. If you choose not to convert your inventory data by specifying the `noatp` option of the migration script, you can convert to ATP at a later time by running the `migrateATP` script as described in the sections below.

Regardless of whether you choose to migrate using the ATP option or not, the database migration script sets up the following tables for you, for the PRODUCT and ITEM entries:

- BASEITEM
- ITEMSPC
- ITEMVERSN
- STOREITEM
- VERSIONSPC

Note that this setup is only done once at migration time. If you add new PRODUCT and ITEM entries after running the database migration script (`migratedb`) you need to ensure that before you run the `migrateATP` script, the above tables are properly set up for any new PRODUCT and ITEM entries that have been added since you have run the migration script. If you use the WebSphere Commerce 5.4 tools to add new products and items, the entries will be set up properly for you.

If you use the mass loader (`massload.xml`) to populate the ITEM and PRODUCT entries, you will also need to set up those tables (BASEITEM, ITEMSPC, ITEMVERSN, STOREITEM, and VERSIONSPC) prior to running the script. For information on how to use the mass loader, refer to the WebSphere Commerce 5.4 online help information on using the Loader, as well as the descriptions of these tables.

The `migrateATP` script creates the following tables:

- For each product, it creates DISTARRANG. It sets the ENDDATE to a large value (59 years) and the STARTDATE to the current date.
- For each item, it creates RADETAIL, RECEIPT, RCTAVAIL, and ITEMFFMCTR.
- For each package, it creates DISTARRANG, RADETAIL, RECEIPT, RCTAVAIL and ITEMFFMCTR. (Treats it as if each parent is a parent product and each item is a child item.)

It also:

- Fills in the TRADING_ID field of all order items to use the store's default contract.
- Generates a script to convert all orders and order items whose status is C to a status of S.

For DB2 databases

If you are using a DB2 database, use the following procedure to run the `migrateATP` script:

___ 1. Open a command prompt window.

___ 2. If your are running WebSphere Application Server as the root user, switch to the DB2 instance user ID for your database (for example, db2inst1):

```
su - db2_instance_user_id
```

If your are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

```
su - WAS_user
```

__ 3. Switch to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example, /usr/WebSphere/CommerceServer/bin.

__ 4. Run the `migrateATP` migration script for DB2 as follows:

```
./migrateATP.db2.sh db_name db_userID db_userID_password db2insthome
```

where:
- *db_name* is the Commerce Suite 5.1 database that you are migrating (for example, `mall`).
- *db_userID* is the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, `mypasswd`).
- *db2insthome* is the home directory for your DB2 instance user ID (for example, /home/db2inst1/sqllib).

For example, to convert your inventory data to the new ATP representation, you can use the following command:

```
migrateATP.db2.sh mall mydbuser mypasswd /home/db2inst1/sqllib
```

__ 5. Check the summary section in the `ATPmigrate.log` which is generated in the /usr/WebSphere/CommerceServer/logs/migration directory to see if any errors or warnings have occurred. If no errors have occurred, you can proceed, otherwise you need to address any error condition and rerun the script before proceeding.

For Oracle databases

If you are using an Oracle database, use the following procedure to run the `migrateATP` script:

__ 1. Open a Command Prompt window

__ 2. If your are running WebSphere Application Server as the root user, switch to the Oracle user ID for your system (for example, `oracle`):

```
su - oracle_user_id
```

If your are running WebSphere Application Server as the non-root user, switch to the WebSphere Application Server user ID (for example, `wasuser`):

```
su - WAS_user
```

__ 3. Traverse to the `bin` subdirectory where WebSphere Commerce 5.4 is installed. For example, /usr/WebSphere/CommerceServer/bin.

__ 4. Run the `migrateATP` migration script for Oracle as follows:

```
./migrateATP.oracle.sh db_name db_userID db_userID_password
  host_name port_number
```

where:
- *db_name* is the Commerce Suite 5.1 database that you are migrating (for example, `mall`).

- *db_userID* is the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, `mydbuser`).
- *db_userID_password* is the password for the user ID to connect to the Commerce Suite 5.1 database that you are migrating (for example, `mypasswd`).
- *host_name* is the fully qualified host name of your machine.
- *port_number* is the Oracle listener port number. Typically, this is port 1521 by default.

For example, to convert your inventory data to the new ATP representation, you can use the following command:

```
./migrateATP.oracle.sh mall mydbuser mypasswd myhost.montreal.ca 1521
```

__ 5. Check the summary section in the `ATPmigrate.log` which is generated in the `/usr/WebSphere/CommerceServer/schema/oracle/migration` directory to see if any errors or warnings have occurred. If no errors have occurred, you can proceed, otherwise you need to address any error condition and rerun the script before proceeding.

Appendix E. Database schema changes

This appendix summarizes the database schema changes between Commerce Suite 5.1 and WebSphere Commerce 5.4. For information on new tables that have been introduced in this release, see the *What's New In IBM WebSphere Commerce, 5.4* document.

For detailed information on the use and the description of the database tables, see the WebSphere Commerce 5.4 online help description of the database tables.

Note: All primary key index names, unique key index names, and foreign key constraint names have changed relative to Commerce Suite 5.1, and are not listed in this table. In WebSphere Commerce 5.4, primary key index names and unique key index names are system-generated. In addition, foreign key constraints have been renamed. For example, for the `CALMETHOD` table, the following table summarizes the changes:

Table 12. CALMETHOD database table

Item	Commerce Suite 5.1 name	WebSphere Commerce 5.4 system-generated name
Primary key index	P_CALMETHOD	SQL010926024834880
Unique key index	UI_CALMETHOD	I0000055
Foreign key constraint	F_CALMETHOD1	F_164
	F_CALMETHOD2	F_163

Database schema changes

The following table lists major changes to the Commerce Suite 5.1 schema.

Table 13. Changed Tables in WebSphere Commerce 5.4

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	Column	Table	Column	
A				
		ACACGPDESC		new table
		ACACTACTGP		new table
		ACACTDESC		new table
		ACACTGRP		new table
		ACACTION		new table
		ACATTR		new table
		ACATTRDESC		new table
ACCCMDGRP				deprecated table
ACCCMDTYPE				deprecated table
ACCCUSTEXC				deprecated table
		ACCLOGMAIN		new table
		ACCLOGSUB		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
ACCMBRGRP				deprecated table
		ACCOUNT		new table
		ACORGPOL		new table
		ACPOLDESC		new table
		ACPOLICY		new table
		ACRELATION		new table
		ACRELDESC		new table
		ACRELGRP		new table
		ACRESACT		new table
		ACRESATREL		new table
		ACRESCGRY		new table
		ACRESGPDES		new table
		ACRESGPRES		new table
		ACRESGRP		new table
		ACRESMEMRL		new table
		ACRESPRIM		new table
		ACRESREL		new table
		ACRLGPDESC		new table
		APRVSTATUS		new table
		ATTACHMENT		new table
		ATTACHUSG		new table
ATTRIBUTE		ATTRIBUTE	USAGE	new column
ATTRVALUE		ATTRVALUE	OPERATOR_ID	new column
		AUCPAYINFO		new table
AUCTION		AUCTION	AUCMODE	new columns
			AUCPAYINFO_ID	
			DURLENGTH	
			INVRSRVID	
			PRICELIMIT	
			QUANTLIMIT	
AUCTIONLOG		AUCTIONLOG	AUCMODE	new columns
			AUCPAYINFO_ID	
			DURLENGTH	
			INVRSRVID	
			PRICELIMIT	
			QUANTLIMIT	
AUTOBID		AUTOBID	AUCPAYINFO_ID	new column
AUTOBIDLOG		AUTOBIDLOG	AUCPAYINFO_ID	new column

B

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		BASEITEM		new table
		BASEITMDSC		new table
BID		BID	AUCPAYINFO_ID	new columns
			BIDMODE	
			INITPRICE	
			LIMITPRICE	
BIDLOG		BIDLOG	AUCPAYINFO_ID	new columns
			BIDMODE	
			INITPRICE	
			LIMITPRICE	
BIDPAYMENT		BIDPAYMENT	AUCPAYINFO_ID	new column
		BKORDALLOC		new table
		BKORDITEM		new table
BUSPROF		BUSPROF	REQUISITIONERID	new column
		BUYERPO		new table
		BUYERPOTYP		new table
		BUYSUPMAP		new table
		BUYSUPSEC		new table
		BZSRVCFG		new table
		BZSVCCFG		new table
		BZSVCSTA		new table
C				
CACHLOG	CACHASHVALUE	CACHLOG	CACHASHVALUE	changed column
	Datatype: VARCHAR(64)		Datatype: VARCHAR(254)	
CALCODE	CODE	CALCODE	CODE	changed column
	Null Option: NULL		Null Option: NOT NULL	
			DESCRIPTION	new columns
			DISPLAYLEVEL	
			ENDDATE	
			FLAGS	
			PRECEDENCE	
			STARTDATE	
		CALCODEMGP		new table
CALRANGE		CALRANGE	MARKFORDELETE	new column

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
CAMPAIGN	ENDDATE	CAMPAIGN		dropped columns
	LASTDEPLOY			
	LASTDEPLOYEDBY			
	MATYPE_ID			
	STARTDATE			
	USERSTATUS			
			STATUS	new columns
			TYPE	
		CATALGDESC		new view
CATALO		CATALO	TPCLEVEL	new column
		CATCLSFCOD		new table
		CATCONFINF		new table
CATEGORY				dropped table
		CATEGRYREL		new view
CATENCALCD		CATENCALCD	TRADING_ID	new column
	CONTRACT_ID			dropped column
CATENTDESC		CATENTDESC	KEYWORD	new column
CATENTRY		CATENTRY	BASEITEM_ID	new columns
			ITEMSPC_ID	
			STATE	
CATGPCALCD		CATGPCALCD	TRADING_ID	new column
	CONTRACT_ID			dropped column
CATGPENREL		CATGPENREL	LASTUPDATE	new column
CATGRPATTR		CATGRPATTR	SEQUENCE	new column
CATGRPDESC		CATGRPDESC	KEYWORD	new column
		CATGRPPS		new table
CATGRPREL		CATGRPREL	LASTUPDATE	new column
		CATGRPTPC		new table
		CATPRDREL		new view
CATTOGRP		CATTOGRP	LASTUPDATE	new columns
			SEQUENCE	
		CHARGETYPE		new table
		CHKARRANG		new table
		CHKCMD		new table
		CHRGTYPDSC		new table
		CLASIFCODE		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
CLEANCONF	CONDITION	CLEANCONF		dropped columns
	TABNAME			
			OBJECTNAME	new columns
			SEQUENCE	
			STATEMENT	
	TYPE		TYPE	changed columns
	Null Option: NULL		Null Option: NOT NULL	
	PK: No		PK: Yes	
	NAMEARGg		NAMEARG	
	Null Option: NULL		Null Option: NOT NULL	
	DAYSARG		DAYSARG	
	Null Option: NULL		Null Option: NOT NULL	
		CLSFCODEDS		new table
CMPGNINTV				dropped table
CMPGNRV				dropped table
		CNTRDISPLY		new table
		CNTRNAME		new table
CONTRACT		CONTRACT	COMMENTS	new columns
			MAJORVERSION	
			MARKFORDELETE	
			MINORVERSION	
			ORIGIN	
			TIMEACTIVATED	
			TIMEAPPROVED	
			TIMECREATED	
			TIMEDEPLOYED	
			TIMEUPDATED	
			USAGE	
	STORE_ID			dropped column
	NAME		NAME	changed columns
	Datatype: VARCHAR(254)		Datatype: VARCHAR(200)	
			Null Option: NOT NULL	
	Null Option: NULL			
	MEMBER_ID		MEMBER_ID	
	Null Option: NULL		Null Option: NOT NULL	
CPENDORDER	MEMBER_ID	CPENDORDER	MEMBER_ID	changed column
	PK: No		PK: Yes	

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
CPGNLOG	INITIATIVE_ID Null Option: NULL PK: No	CPGNLOG	INITIATIVE_ID Null Option: NOT NULL PK: Yes	changed column
		CPITMAP		new table
		CPOFFER		new table
		CPPMN		new table
		CPPMNDESC		new table
		CPPMNDISC		new table
		CPPMNORD		new table
		CPPMNPROD		new table
		CPPMNVAL		new table
		CPWALLET		new table
		CREDITLINE		new table
CURFMTDESC		CURFMTDESC	NUMBRUSG_ID	new column
CURFORMAT		CURFORMAT	NUMBRUSG_ID	new columns
			MINAPPROVEAMOUNT	
D				
DISPCGPREL		DISPCGPREL	LANGUAGE_ID	new column
DISPENTREL		DISPENTREL	LANGUAGE_ID	new column
		DISTARRANG		new table
E				
		EMSPOT		new table
		ENUMDESC		new table
F				
FFMCENTER		FFMCENTER	DEFAULTSHIPOFFSET	new columns
			MARKFORDELET	
		FLCOMPOSE		new table
		FLDOMNDESC		new table
		FLINSTANCE		new table
		FLOW		new table
		FLOWADMIN		new table
		FLOWDESC		new table
		FLOWDOMAIN		new table
		FLOWTYPE		new table
		FLSTATEDCT		new table
		FLSTATEGP		new table
		FLSTATEREL		new table
		FLSTDCTDSC		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		FLSTGPDSC		new table
		FLTRANSDSC		new table
		FLTRANSITN		new table
		FLTYPEDESC		new table
H				
		HISTOATTR		new table
		HISTOFREQ		new table
		HISTONVP		new table
I				
ICEXPLDESC	NAME	ICEXPLDESC	NAME	changed column
	Datatype: VARCHAR(64)		Datatype: VARCHAR(254)	
ICEXPLFEAT	COLUMNNAME	ICEXPLFEAT	COLUMNNAME	changed column
	Datatype:CHAR		Datatype: VARCHAR(254)	
INITIATIVE		INITIATIVE	CAMPAIGN_ID	new columns
			RESULTTYPE	
			RULE	
			SELLTYPE	
			STATUS	
			TYPE	
	DEPLOYABLE			dropped columns
	ENDDATE			
	LASTDEPLOY			
	LASTDEPLOYEDBY			
	MATYPE_ID			
	PROFILE			
	RULEVALUE_ID			
	STARTDATE			
	USERSTATUS			
INTVMPE				dropped table
		INTVSCHED		new table
INTVSGMT				dropped table
		INVADJCODE		new table
		INVADJDESC		new table
		INVADJUST		new table
		INVOICE		new table
		INVRESERVE		new table
		INVRSRVDSC		new table
		INVRSRVTYP		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		ITEMFFMCTR		new table
		ITEMSPC		new table
		ITEMTYPE		new table
		ITEMVERSN		new table
J				
JURST		JURST	MARKFORDELETE	new column
JURSTGROUP		JURSTGROUP	MARKFORDELETE	new column
L				
LANGUAGE		LANGUAGE	MIMECHARSET	new column
LISTPRICE				dropped table
		LMEVENTMAP		new table
		LMSERVER		new table
		LPPOPURAMT		new table
M				
MAFAMILY				dropped table
		MANIFEST		new table
MATYPE				dropped table
		MBRATTR		new table
		MBRATTRVAL		new table
MBRGRP		MBRGRP	LASTUPDATE	new columns
			LASTUPDATEDBY	
		MBRGRPCOND		new table
MBRGRPMBR		MBRGRPMBR	EXCLUDE	new column
MBRGRPTYPE		MBRGRPTYPE	PROPERTIES	new column
		MBRREL		new table
		MBRROLE		new table
MEMBER		MEMBER	STATE	new column
		MLTIME		new table
MPE				dropped table
MPETYPE				dropped table
N				
		NUMBRUSG		new table
		NUMBRUSGDS		new table
O				
OFFER	IDENTIFIER	OFFER	IDENTIFIER	changed column
	Null Option: NOT NULL		Null Option: NULL	
		OICOMPLIST		new table
		OPERATOR		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	Column	Table	Column	
		OPERATRDSC		new table
		ORCPMAP		new table
ORDADJUST		ORDADJUST	DISPLAYLEVEL	new column
		ORDCALCD		new table
		ORDCHNLTYP		new table
ORDERITEMS		ORDERITEMS	ALLOCADDRESS_ID	new columns
			ALLOCATIONGROUP	
			ALLOCFFMC_ID	
			ALLOCQUANTITY	
			CONFIGURATIONID	
			CORRELATIONGROUP	
			ESTAVAILTIME	
			FULFILLMENTSTATUS	
			INVENTORYSTATUS	
			ITEMSPC_ID	
			LASTALLOCUPDATE	
			LINEITEMTYPE	
			NEEDEDQUANTITY	
			ORDRELEASENUM	
			OUTPUTQ_ID	
			PROMISEDAVAILTIME	
			SHIPPINGOFFSET	
			TERMCOND_ID	
			TIMERELEASED	
			TIMESHIPPED	
			TRADING_ID	
	CONTRACT_ID			dropped column
		ORDERMSG		new table
ORDERS		ORDERS	ORDCHNLTYP_ID	new columns
			ORGENTITY_ID	
			PROVIDERORDERNUM	
			SHIPASCOMPLETE	

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	Column	Table	Column	
ORDICALCD		ORDICALCD	CALFLAGS	new columns
			CALPARMAMT	
			CALPARMTYPE	
			ORDICALCD_ID	
	CALCODE_ID		CALCODE_ID	changed columns
	PK: Yes		PK: No	
	ORDERITEMS_ID		ORDERITEMS_ID	
	PK: Yes		PK: No	
		ORDIMEEXTN		new table
		ORDIOFFER		new table
		ORDIPROF		new table
		ORDITRD		new table
		ORDMEEXTN		new table
ORDOPTIONS	NOTIFYMERCHANT	ORDOPTIONS	NOTIFYMERCHANT	changed columns
	Datatype: SMALLINT		Datatype: INTEGER	
	NOTIFYSHOPPER		NOTIFYSHOPPER	
	Datatype: SMALLINT		Datatype: INTEGER	

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
ORDPAYMTHD		ORDPAYMTHD	ACCOUNT_ID	new columns
			ACTUALAMOUNT	
			BIGINTFIELD1	
			BIGINTFIELD2	
			BIGINTFIELD3	
			BUYERPO_ID	
			CHARGEAMOUNT	
			CHARGEAMTCURR	
			CHARGETIME	
			CREDITLINE_ID	
			DECIMALFIELD1	
			DECIMALFIELD2	
			DECIMALFIELD3	
			PAYSUMMARY_ID	
			POLICY_ID	
			REFUNDNUMBER	
			RMA_ID	
			STATUS	
			STRINGFIELD1	
			STRINGFIELD2	
			STRINGFIELD3	
			STRINGFIELD4	
			TRADING_ID	
			XMLDATA	
		ORDPICKHST		new table
		ORDRELEASE		new table
		ORDSHIPHST		new table
ORDTAX		ORDTAX	LASTUPDATE	new column
		ORGCODE		new table
ORGENTITY		ORGENTITY	MEMBER_ID	deprecated column
		OUTPUTQ		new table
		OUTPUTQDSC		new table
P				
		PARTICIPNT		new table
		PARTROLE		new table
		PARTROLEDS		new table
		PATTRIBUTE		new table
		PATTRPROD		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		PATTRVALUE		new table
PAYMTHD		PAYMTHD		deprecated table
PAYMTHDDSC		PAYMTHDDSC		deprecated table
PATMTHDSUP		PAYMTHDSUP		deprecated table
PAYSTATUS	PENDING	PAYSTATUS	PENDING	changed column
	Datatype: SMALLINT		Datatype: INTEGER	
		PAYSUMMARY		new table
		PICKBATCH		new table
PKGATTR	LANGUAGE_ID	PKGATTR	LANGUAGE_ID	changed table
	PK: No		PK: Yes	
PKGATTRVAL	LANGUAGE_ID	PKGATTRVAL	LANGUAGE_ID	changed table
	Null Option: NULL		Null Option: NOT NULL	
	PK: No		PK: Yes	
PKGITEMREL		PKGITEMREL		depracated table
		PLCYACCDSC		new table
		PLCYACCLCK		new table
		PLCYACCT		new table
		PLCYLCKDSC		new table
		PLCYPASSWD		new table
		PLCYPWDDSC		new table
		PLCYTYCMIF		new table
		PLCYTYPDSC		new table
		POLICY		new table
		POLICYCMD		new table
		POLICYDESC		new table
		POLICYTC		new table
		POLICYTYPE		new table
		PRATRSTATR		new view
		PRCOFFRPRC		new view
		PRCEOFFPRD		new view
		PRDATRAVAL		new view
		PRICE		new view
		PRICEDESC		new view
		PROCBUYPRF		new table
		PROCMSGVW		new table
		PROCPROTCL		new table
		PROCSYS		new table
		PRODATR		new view

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		PRODDSTATR		new view
		PRODSETDSC		new table
PRODUCT				dropped summary table
		PRODUCTSET		new table
		PRSETCEREL		new table
		PURCHASELT		new table
PVCBINDING		PVCBINDING	PVCBINDING_ID	new column
	PVCSESSION_ID		PVCSESSION_ID	changed columns
	PK: Yes		PK: No	
	USERS_ID		USERS_ID	
	PK: Yes		PK: No	
PVCBUFFER		PVCBUFFER	PARAMETERS	deprecated column
			PARAMETERS2K	new column
PVCDEVMDL	MODELNAME	PVCDEVMDL	MODELNAME	changed column
	Null Option: NOT NULL		Null Option: NULL	
			DEVMDLNAME	new column
			MODELNAME	deprecated column
PVCDEVSPEC	SESSIONTYPE	PVCDEVSPEC	SESSIONTYPE	changed column deprecated column
	Null Option: NOT NULL		Null Option: NULL	
Q				
QTYFMTDESC		QTYFMTDESC	NUMBRUSG_ID	new column
QTYFORMAT		QTYFORMAT	NUMBRUSG_ID	new column
		QTYUNITMAP		new table
R				
		RA		new table
		RABACKALLO		new table
		RADETAIL		new table
		RCPTAVAIL		new table
		RECEIPT		new table
		REFUNDMTHD		new table
RICHATTR				table has been redefined
RICHATTRCG				new table
RICHATTRCATGP				dropped summary table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		RFQ		new table
		RFQPROD		new table
		RFQRSP		new table
		RFQRSPPROD		new table
		RFQRSPTCRL		new table
		RLDISCOUNT		new table
		RMA		new table
		RMAAUTHLOG		new table
		RMACHARGE		new table
		RMAIADJCRD		new table
		RMAIDNYRSN		new table
		RMAITEM		new table
		RMAITEMCMP		new table
		RMATAX		new table
		ROLE		new table
		RTNDNYDESC		new table
		RTNDNYRSN		new table
		RTNDSPCODE		new table
		RTNDSPDESC		new table
		RTNRCPTDSP		new table
		RTNREASON		new table
		RTNRECEIPT		new table
		RTNRSNDESC		new table
S				
		SCHCMD		new table
SCHCONFIG		SCHCONFIG	INTERFACENAME	new columns
			STOREENT_ID	
	SCCAPPTYPE		SCCAPPTYPE	changed columns
	Datatype: VARCHAR		Datatype: CHAR	
	SCCPATHINFO		SCCPATHINFO	
	Datatype: VARCHAR(128)		Datatype: VARCHAR(254)	
	SCCQUERY		SCCQUERY	
	Datatype: VARCHAR		Datatype: LONG	
SEGMENT				dropped table
		SCHACTIVE		new table
SHIPMODE		SHIPMODE	MARKFORDELETE	new column
		SHPARJURGP		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
SHPARRANGE		SHPARRANGE	FLAGS	new column
	SHIPMODE_ID		SHIPMODE_ID	changed column
	Null Option: NOT NULL		Null Option: NULL	
SHPJCRULE	FFMCENTER_ID	SHPJCRULE	FFMCENTER_ID	changed columns
	Null Option: NOT NULL		Null Option: NULL	
	JURSTGROUP_ID		JURSTGROUP_ID	
	Null Option: NOT NULL		Null Option: NOT NULL	
	SHIPMODE_ID		SHIPMODE_ID	
	Null Option: NOT NULL		Null Option: NOT NULL	
STDPRICE				new table
STENCALUSG		STENCALUSG	CALMETHOD_ID_APP	new columns
			CALMETHOD_ID_FIN	
			CALMETHOD_ID_INI	
			CALMETHOD_ID_SUM	
			SEQUENCE	
			USAGEFLAGS	
		STGUINDTAB		new table
STORE		STORE	ALLOCATIONGOODFOR	new columns
			AVSACCEPTCODES	
			BOPMPADFACTOR	
			DEFAULTBOOFFSET	
			FFMCSELECTIONFLAGS	
			MAXBOOFFSET	
			PRICEREFFLAGS	
			REJECTEDORDEXPIRY	
			RMAGOODFOR	
			RTNFFMCTR_ID	
			STORETYPE	
STORECAT		STORECAT	LASTUPDATE	new columns
			MASTERCATALOG	
		STORECNTR		new table
STOREENT		STOREENT	SETCCURR	new column
STOREINV		STOREINV		Changed from a summary table to a view
		STOREITEM		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		STORITMFFC		new table
T				
TAXCGRY		TAXCGRY	MARKFORDELETE	new column
TAXJCRULE		TAXJCRULE	TAXJCRULE_ID	new column
	CALRULE_ID		CALRULE_ID	changed columns
	PK: Yes		PK: No	
	FFMCENTER_ID		FFMCENTER_ID	
	Null Option: NOT NULL		Null Option: NULL	
	PK: Yes		PK: No	
	JURSTGROUP_ID		JURSTGROUP_ID	
	Null Option: NOT NULL		Null Option: NULL	
	PK: Yes		PK: No	
TAXTYPE		TAXTYPE	SEQUENCE	deprecated column
		TCDESC		new table
		TCPITMAP		new table
		TCSUBTYPDS		new table
		TCSUBTYPE		new table
		TCTYPE		new table
		TERMCOND		new table
		TFALGOPOL		new table
		TFALGOREG		new table
		TFALGOTYPE		new table
		TFALGPOLDS		new table
		TFALGTYPDS		new table
		TFALREGDSC		new table
		TFDOMAIN		new table
		TFDOMDSC		new table
		TFEXTENG		new table
		TFSBDOMAIN		new table
		TFSBDOMDSC		new table
		TFTRADENG		new table
		TFTRENGDSC		new table
		TMPBOLIST		new table
		TMPCMPLIST		new table
		TMPPROCID		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		TMPFFCLIST		new table
		TMPRADTL		new table
		TMPRCTLIST		new table
		TORCPMAP		new table
TRADEPOSCN		TRADEPOSCN	PRODUCTSET_ID	new columns
			PRECEDENCE	
			MARKFORDELETE	
			TYPE	
	NAME		NAME	changed column
	Null Option: NULL		Null Option: NOT NULL	
		TRADING		new table
		TRDATTACH		new table
		TRDDEPAMT		new table
		TRDDESC		new table
		TRDPURAMT		new table
		TRDREFAMT		new table
		TRDTYPE		new table
		TRDTYPEDSC		new table
U				
USERPVCDEV		USERPVCDEV	DEVICEIDENTIFIER	deprecated column
			PVCSESSION_ID	new column
USERREG		USERREG	PLCYACCT_ID	new columns
			TIMEOUT	
			PASSWORDRETRIES	
			SALT	
			PASSWORDCREATION	
			PASSWORDINVALID	
USRTRAFFIC		USRTRAFFIC	SESSIONID	new column
V				
		VENDOR		new table
		VENDORDESC		new table
		VERSIONSPC		new table
W				
		WCCATDESC		new view
		WCCATEGORY		new view
		WCCATRTCAT		new view
		WCCATSTORS		new view
		WCMLANG		new table

Table 13. Changed Tables in WebSphere Commerce 5.4 (continued)

WebSphere Commerce Suite 5.1		WebSphere Commerce 5.4		Comment
Table	**Column**	**Table**	**Column**	
		WCPRDCTPID		new view
		WCPRDPRICE		new view
		WCPRODCATP		new view
		WCPRODDESC		new view
		WCPRODSHIP		new view
		WCPRODUCT		new view
		WCSDTNRYDS		new table
Z				
ZIPCODE				dropped table

Appendix F. Changed programming interfaces

This appendix lists the changes to programming interfaces such as commands, class names and so on in Commerce Suite 5.1 that are either changed or discontinued in WebSphere Commerce 5.4. For information on new interfaces that have been introduced in this release, see the *What's New In IBM WebSphere Commerce, 5.4* document.

For detailed information on the use and the syntax of any of the changed interfaces, see the WebSphere Commerce 5.4 online help.

Deprecated commands

Note that the following Commerce Suite 5.1 commands have been deprecated in WebSphere Commerce 5.4 and will not be supported in future releases of WebSphere Commerce.

- `EntityAdmin`

  This command was documented in the online help in Commerce Suite 5.1, but is no longer documented in the WebSphere Commerce 5.4 online help. In addition, this command used to support the following entities:

  - `entity=MemberGroupType` (maps to MBRGRPTYPE table)
  - `entity=MemberGroup` (maps to MBRGRP table)
  - `entity=MemberGroupMember` (maps to MBRGRPMBR table)
  - `entity=MemberGroupUsage` (maps to MBRGRPUSG table)
  - `entity=`**`AccessControlMemberGroup`** (maps to ACCMBRGRP table)
  - `entity=`**`AccessControlCommandGroup`** (maps to ACCCMDGRP table)
  - `entity=`**`AccessControlCustomerCommandExclusion`** (maps to ACCCUSTEXC table)

  Note: The last three entities above (in bold) were used to manage the Commerce Suite 5.1 access control tables, that are no longer used in WebSphere Commerce 5.4. So, even though the command can still manipulate those tables in WebSphere Commerce 5.4, the access control runtime is not affected by data in those tables.

  For more information on access control, see "Access control" on page 9 and Chapter 8, "Access control subsystem considerations," on page 137.

- `HTTPCommandContext`

  The `HTTPCommandContext` interface has been deprecated in WebSphere Commerce 5.4 as the same function is provided in the `CommandContext` command. You need to use the `CommandContext` command instead of `HTTPCommandContext` in your code or you will receive runtime `ClassCastException` exception.

  If you are currently using `HTTPCommandContext`, change your code in the following way:

  1. To get the `HttpServletRequest` object:

     ```
     com.ibm.commerce.webcontroller.HttpControllerRequestObject req =
     (com.ibm.commerce.webcontroller.HttpControllerRequestObject)commandContext.getRequest();
     HttpServletRequest httpRequest = req.getHttpRequest();
     ```

  2. To get the `HttpServletResponse` object:

     ```
     HttpServletResponse httpResponse = (HttpServletResponse)commandContext.getResponse();
     ```

  3. To check whether a request comes from the browser:

```
if (commandContext.getRequest() instanceof
    com.ibm.commerce.webcontroller.HttpControllerRequestObject)
```

or

```
if (commandContext.getDeviceFormatTypeId().equals(new Integer(-1)))
```

Changed commands

Product Advisor

Product Advisor package names have changed from WebSphere Commerce Suite
5.1. If you are using Product Advisor, see "Product Advisor migration
considerations" on page 129 for information on these package names.

UserRegistrationAddCmd and UserRegistrationUpdateCmd

In WebSphere Commerce Suite 5.1, in the
`com.ibm.commerce.usermanagement.command` package, the `UserRegistrationAddCmd`
and `UserRegistrationUpdateCmd` commands extend the `UserRegistrationCmd`
command which extends the `AddressBaseCmd` command.

In WebSphere Commerce 5.4, the `UserRegistrationCmd` command no longer
extends the `AddressBaseCmd` command. So if there are methods that you are using
that you inherit from `AddressBaseCmd`, you will need to rewrite your code as
indicated in bold below:

For example:
```
if ( isGuest ){
 UserRegistrationAddCmd userRegAdd = (UserRegistrationAddCmd)
   CommandFactory.createCommand(UserRegistrationAddCmd.NAME,getStoreId());
  userRegAdd.setCommandContext(getCommandContext());
  userRegAdd.setRequestProperties(register_Prop);
 userRegAdd.execute();
 newBillingAddress=userRegAdd.getAddressId(); <===in 51
 //should change to the following in 54
String userId = userRegAdd.getWorkingUserId();
   try
{
 AddressAccessBean abBillingAddress =
     new AddressAccessBean().findSelfAddressByMember(new Long(userId));
 newBillingAddress=abBillingAddress.getAddressId() ;
}
catch(Exception e)
{
 //User does not have a self address
}
 ECTrace.trace(ECTraceIdentifiers.COMPONENT_USER, CLASSNAME, METHODNAME,
     "Successfully created a new user..");

 }
 else {
   UserRegistrationUpdateCmd userRegUpdate = (UserRegistrationUpdateCmd)
         CommandFactory.createCommand(UserRegistrationUpdateCmd.NAME,getStoreId());
    userRegUpdate.setCommandContext(getCommandContext());
    userRegUpdate.setRequestProperties(register_Prop);
   userRegUpdate.execute();
   newBillingAddress = userRegUpdate.getAddressId();<==in 51
  //should change to the following in 54
String userId = userRegAdd.getWorkingUserId();
try
{
 AddressAccessBean abBillingAddress =
    new AddressAccessBean().findSelfAddressByMember(new Long(userId));
 newBillingAddress=abBillingAddress.getAddressId() ;
}
catch(Exception e)
{
 //Use does not have a self address
}
```

```
                    ECTrace.trace(ECTraceIdentifiers.COMPONENT_USER, CLASSNAME, METHODNAME,
                        "Successfully updated a new user..");
                    }
```

AddressAdd command

In WebSphere Commerce 5.4, the `AddressAdd` command contains additional
function to create an address for any specified member (including the
organization). Thus it takes `memberId` as a parameter. If the `memberId` is specified, it
will create the address for that member. If is not specified, it will create address for
the current user.

Note that for Commerce Suite 5.1, the `AddressAdd` command did not have `memberId`
as parameter. It only creates an address for the current user. So for Commerce
Suite 5.1, the `memberId` parameter is ignored even it is specified.

Changed programming interfaces

The tables in the following sections list changes to the programming interfaces
contained in the following packages:

- WCS_Order
- WCS_Catalog
- WCS_User
- WCS_Databean
- WCS_EJB

In these tables, the complete name for all classes begin with `com.ibm.commerce..`
For example, the complete name of `taxation.commands.GetDisplayTaxesCmdImpl` is
`com.ibm.commerce.taxation.commands.GetDisplayTaxesCmdImpl`. The
`com.ibm.commerce.` has been dropped from the names in the following table for
brevity

WCS_Order

The following table summarizes changes to classes related to the Order subsystem:

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
taxation.commands.GetDisplayTaxesCmdImpl	BigDecimal BIG_DECIMAL_ZERO;	public void setDisplayOnly(boolean);
	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public void setUOM(String);
taxation.commands.ApplyOrderTaxesCmd	public static final String defaultCommandClassName;	
taxation.commands.ResolveTaxJurisdictionCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
fulfillment.commands.InventoryBaseCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
fulfillment.commands.ApplyOrderShipping ChargesCmd	public static final String defaultCommandClassName;	
fulfillment.commands.UpdateInventoryFor OrderItemsCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
price.commands.GetBaseUnitPriceCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		static Class classcomibm$commerce $price$commands $GetBaseUnitPrice CmdImpl;
		static Class class$(String);
price.commands.GetBaseUnitPriceCmdImpl $QualifyingOfferInfo	GetBaseUnitPriceCmdImpl this$0;	
price.commands.ApplyOrderAdjustmentsCmd	public static final String defaultCommandClassName;	
price.commands.SetCurrencyPreferenceCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		static Class classcomibm$commerce $price$commands $SetCurrencyPreference CmdImpl;
order.utils.GetCalculationUsagesCmdImpl $PseudoOrderItemAccessBean	GetCalculationUsagesCmdImpl this$0;	
order.utils.GetCalculationUsagesCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.utils.OrderConstants	public static final String ORDER_BID;	public static final String ORDER_BACKORDERED;
		public static final String ORDER_PAYMENT_ AUTH_REVIEW;
order.utils.GetShippingChargesCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public void setCurrency(String);
order.utils.ResolveOrdersCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		static Class class$java$lang$Long;
order.utils.ResolveJurisdictionCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.utils.ApplyCalculationUsagesCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.utils.CalculationCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.utils.ResolveOrderItemsCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		static Class class$java$lang$Long;
order.commands.CheckOrderCopyCmdImpl	public void checkParameters() throws ECException;	

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
order.commands.OrderDisplayCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.OrderProfileUpdateCmdImpl	public void checkParameters() throws ECException;	
order.commands.OrderListCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.CheckOrderTemplateCmdImpl	public void checkParameters() throws ECException;	
order.commands.ExtOrderCopyCmdImpl	public void checkParameters();	
order.commands.SetOrderPaymentInfoCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.OrderProcessCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.OrderProcessCmdImpl $OrderExpiredException	OrderProcessCmdImpl this$0;	
order.commands.OrderProcessingHelper $DoubleNVP	OrderProcessingHelper this$0;	
order.commands.AdminOrderCancelCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.ScheduledOrderCancelCmd Impl	public void checkParameters() throws ECException;	public AccessVector getResources() throws ECException;
order.commands.OrderCopyCmdImpl	public void checkParameters() throws ECException;	public AccessVector getResources() throws ECException;
		public void validateParameters() throws ECException;
order.commands.GetOrderPayInfoCmdImpl	public void checkParameters() throws ECException;	
order.commands.SetCurrentPendingOrdersCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.GetCurrentPendingOrdersCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.OrderPrepareCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public void setATPParms (ATPParameters);
order.commands.ScheduledOrderProcessCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public AccessVector getResources() throws ECException;
order.commands.OrderProfile	public void updatePaymentInfo(Hashtable) throws FinderException, CreateException, RemoteException, NamingException, ECApplicationException, ECException;	

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
		public void updatePaymentInfo (Hashtable, CommandContext) throws FinderException, CreateException, RemoteException, NamingException, ECApplicationException, ECException;
order.commands.SetOrderTemplateCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		static Class class$java$lang$String;
order.commands.OrderScheduleCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.TestNotificationCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
order.commands.OrderUnlockCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public AccessVector getResources() throws ECException;

WCS_Catalog

The following table summarizes changes to classes related to the Catalog subsystem:

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
tools.catalog.commands.OfferNotebookUpdate Impl		removed
tools.catalog.commands.ItemNotebookUpdate Impl		removed
tools.catalog.commands.OfferNotebookUpdate		removed
tools.catalog.commands.ItemNotebookUpdate		removed
tools.catalog.beans.ItemListBean		removed
tools.catalog.beans.OfferListBean		removed
tools.catalog.beans.LanguageDataBean		removed
tools.catalog.beans.CalculationCodeSBBean		removed
tools.catalog.beans.OfferDataBean		removed
tools.catalog.beans.ListPriceDataBean		removed
tools.catalog.beans.OfferPriceDataBean		removed
tools.catalog.beans.CatalogListDataBean		removed
tools.catalog.beans.TaxCodeClassificationData Bean		removed
tools.catalog.beans.QtyUnitListData		removed
tools.catalog.beans.ProductSearchData		removed
tools.catalog.beans.AttributeValueDataBean	public void setCommandContext (CommandContext);	
	public TypedProperty getRequestProperties();	public Long getAttributeValueId();

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
	public CommandContext getCommandContext();	public String getDisplaySequence();
	public void setRequestProperties (TypedProperty);	public void setAttributeValueId(Long);
tools.catalog.helpers._CatalogSearch_BaseStub	public Vector findCatalogEntry (ProductSearchData) throws NamingException, SQLException, RemoteException, ObjectNotFoundException;	
		public Vector findCatalogEntry (ProductSearchParameters) throws NamingException, SQLException, RemoteExceptionObjectNotFound Exception;
tools.catalog.helpers._CatalogSearch_Stub	public Vector findCatalogEntry (ProductSearchData) throws NamingException, SQLException, RemoteExceptionObjectNotFound Exception;	
		public Vector findCatalogEntry (ProductSearchParameters) throws NamingExceptionSQLException, RemoteExceptionObjectNotFound Exception;
tools.catalog.helpers.CatalogSearchBean	public static final String findCatalogEntryTypeID_WHERE;	public static final String findItem_WHERE;
		public static final String findProduct_WHERE;
		public void setCatalogEntryType(String);

WCS_User

The following table summarizes changes to classes related to the User subsystem:

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
usermanagement.commands.AddressDelete CmdImpl	public void checkParameters() throws ECException;	
	public boolean isReadyToCallExecute();	
usermanagement.commands.UserRegistration CmdImpl	public void checkParameters() throws ECException;	public boolean isLogonIdUnique(String) throws ECException;
		public void setCity(String);
usermanagement.commands.OrgEntity RegistrationCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
usermanagement.commands.AddressCheck CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
	public boolean isReadyToCallExecute();	
usermanagement.commands.AddressBase CmdImpl	public void checkParameters() throws ECException;	public String getAddress3();
		public AccessVector getResources() throws ECException;
security.commands.LogonCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
		public void updateLockoutInformation(boolean) throws ECException;
security.commands.LDAPAuthenticationCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public String getAuthenticateUserId();
security.commands.DBAuthenticationCmd Impl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public void markPasswordAsTemporary();
security.commands.LogoffCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.ResetPasswordRegister Cmd	public abstract String getErrorCode();	
security.commands.UpdateCredentials CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public void markPasswordTemporary();
security.commands.ResetPasswordCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.VerifyCredentialsCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
		public String getAuthenticateUserId();
security.commands.ResetPasswordGuest CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.MigrateUserEntries CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.SendPasswordNotification CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.ResetPassword AdministratorCmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
security.commands.ResetPasswordRegister CmdImpl	public void checkParameters() throws ECException;	public void validateParameters() throws ECException;
	public String getErrorCode();	

WCS_Databean

The following table summarizes changes to classes related to the Databean subsystem:

Classes	WebSphere Commerce Suite 5.1	WebSphere Commerce 5.4
user.beans.UserRegistrationInputDataBean	public abstract void setUserId(Long);	

Enterprise JavaBeans

The following table summarizes changes to Enterprise JavaBeans classes:

Table 14.

Component	Class names	Methods only in WebSphere Commerce Suite 5.1
Catalog	AttributeBean	public void ejbCreate(Long, Integer) throws CreateExceptionRemoteException; This method has been removed from WebSphere Commerce 5.4.

Table 14. (continued)

Component	Class names	Methods only in WebSphere Commerce Suite 5.1
	AttributeFloatValueHome	public abstract AttributeFloatValue create(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	AttributeHome	public abstract Attribute create(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	AttributeIntegerValueHome	public abstract AttributeIntegerValue create(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	AttributeStringValueHome	public abstract AttributeStringValue create(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	AttributeValueBean	public void ejbCreate(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	AttributeValueHome	public abstract AttributeValue create(Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	CatalogEntryHome	public abstract CatalogEntry create(Long) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	CatalogGroupPageRelationBean	public void ejbCreate(Long, Long, Integer) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	ItemHome	public abstract Item create(Long) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	ProductHome	public abstract Product create(Long) throws CreateExceptionRemoteException;
		This method has been removed from WebSphere Commerce 5.4.
	ProductUserGroupRelationBean	This class file has been removed from WebSphere Commerce 5.4.
		This method has been removed from WebSphere Commerce 5.4.

Table 14. (continued)

Component	Class names	Methods only in WebSphere Commerce Suite 5.1

Order	OfferBean	public Enumeration getContracts() throws FinderExceptionRemoteExceptionNaming Exception;
		
		public Enumeration getContracts(Integer) throws FinderExceptionRemoteExceptionNaming Exception;
		
		This is a design change in WebSphere Commerce 5.4. Contract now cannot be retrieved through an offer.

	OrderOptionBean	All Short java object type has been migrated to Integer object type to be consistent with the database.
Order Status	OrderFulfillmentItemStatusBean	public void ejbCreate(TypedProperty) throws CreateExceptionNamingExceptionRemote ExceptionFinderException; TypedProperty is now changed to a Hashtable in order to remove runtime dependency for EJB object.
	OrderFulfillmentStatusBean	public void ejbCreate(TypedProperty) throws CreateExceptionNamingExceptionRemote ExceptionFinderException; TypedProperty is now changed to a Hashtable in order to remove runtime dependency for EJB object.
	OrderFulfillmentStatusHome	public abstract OrderFulfillmentStatus create(TypedProperty) throws NamingExceptionCreate ExceptionFinder ExceptionRemoteException; TypedProperty is now changed to a Hashtable in order to remove runtime dependency for EJB object
User	MemberBean	public String getDisplayName(); Display name column has been removed from the MEMBER table.
	MemberBeanFinderHelper	public static final String findDistinctStoreOwnersWhereClause; This method was not operational in WebSphere Commerce Suite 5.1 and has been removed from WebSphere Commerce 5.4.

Appendix G. Updates to sample JSPs

In order for aspects of your store to work correctly in WebSphere Commerce 5.4 after migration some of the JSPs need to be modified. For example, to be able to complete shopping flows using Store Services, you need to update `shipaddress.jsp` that was shipped in Commerce Suite 5.1. In addition, to improve logon error messaging, you need to update the `register.jsp` and `account.jsp` JSPs that were shipped in Commerce Suite 5.1.

The updated JSPs are listed here for your reference.

register.jsp

```
<%
//****************************************************************
//*———————————————————————————————————————————————————————————
//* Licensed Materials - Property of IBM
//*
//* 5697-D24
//*
//* (c) Copyright IBM Corp. 2000, 2002
//*
//* US Government Users Restricted Rights - Use, duplication or
//* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
//*
//*———————————————————————————————————————————————————————————
//*
%>

<%@ page language="java" %>
<% // All JSPs requires the first 4 packages for getResource.jsp which is used for multi language support %>
<%@ page import="java.io.*" %>
<%@ page import="java.util.*" %>
<%@ page import="com.ibm.commerce.server.*" %>
<%@ page import="com.ibm.commerce.command.*" %>

<%@ page import="javax.servlet.*" %>
<%@ page import="com.ibm.commerce.catalog.beans.*" %>
<%@ page import="com.ibm.commerce.catalog.objects.*" %>
<%@ page import="com.ibm.commerce.beans.*" %>

<%@ page import="com.ibm.commerce.user.beans.*" %>
<%@ page import="com.ibm.commerce.user.objects.*" %>
<%@ page import="com.ibm.commerce.datatype.*" %>
<%@ page import="com.ibm.commerce.usermanagement.commands.ECUserConstants" %>
<%@ page import="com.ibm.commerce.common.beans.*" %>
<%@ page import="com.ibm.commerce.security.commands.ECSecurityConstants" %>
<%@ include file="getResource.jsp"%>
<%
// JSPHelper provides you with a easy way to retrieve
//            URL parameters when they are encrypted
JSPHelper jhelper = new JSPHelper(request);

String storeId = jhelper.getParameter("storeId");
String catalogId = jhelper.getParameter("catalogId");
String languageId = jhelper.getParameter("langId");
%>

<jsp:useBean id="bnError" class="com.ibm.commerce.beans.ErrorDataBean" scope="page">
<% com.ibm.commerce.beans.DataBeanManager.activate(bnError, request); %>
</jsp:useBean>

<%
String strErrorMessage = null;
String strErrorCode = "";

String strLogonID = null;
String strPassword = null;
String strPasswordVerify = null;
String strLastName = null;
String strFirstName = null;
```

```
TypedProperty hshErrorProperties = bnError.getExceptionData();

if (hshErrorProperties != null)
{
        //We have a registration error.

        strErrorCode = hshErrorProperties.getString(ECConstants.EC_ERROR_CODE, "");
        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_BAD_LOGONID))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE40");
        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_LOGONID_EXISTS))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE41");

        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_BAD_LOGONPASSWORD))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE42");
        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_BAD_LOGONPASSWORDVERIFY))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE43");
        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_PASSWORDS_NOT_SAME))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE44");

        if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_LASTNAME))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE45");
        if (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_MISSING_LOGONPASSWORDVERIFY))
                strErrorMessage = infashiontext.getString("ERROR_MESSAGE46");

        if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMLENGTH_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE21");
        if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMDIGITS_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE22");
        if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMLETTERS_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE23");
        if (strErrorCode.equals(ECSecurityConstants.ERR_USERIDMATCH_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE24");
        if (strErrorCode.equals(ECSecurityConstants.ERR_REUSEOLD_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE25");
        if (strErrorCode.equals(ECSecurityConstants.ERR_MAXCONSECUTIVECHAR_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE26");
        if (strErrorCode.equals(ECSecurityConstants.ERR_MAXINTANCECHAR_PASSWORD))
                strErrorMessage = infashiontext.getString("PASS_ERROR_MESSAGE27");
        //Redisplay what was entered when the
        //invalid entry was submitted.
        strLogonID = jhelper.htmlTextEncoder(jhelper.getParameter(ECUserConstants.EC_UREG_LOGONID));
        strPassword = jhelper.getParameter(ECUserConstants.EC_UREG_LOGONPASSWORD);
        strPasswordVerify = jhelper.getParameter(ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY);
        strLastName = jhelper.htmlTextEncoder(jhelper.getParameter(ECUserConstants.EC_ADDR_LASTNAME));
        strFirstName = jhelper.htmlTextEncoder(jhelper.getParameter(ECUserConstants.EC_ADDR_FIRSTNAME));
}
else
{
        //Form is loading under regular condition.
        //Initialize all fields to empty.

        strLogonID = "";
        strPassword = "";
        strPasswordVerify = "";
        strLastName = "";
        strFirstName = "";
}
%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
<head>
<title><%=infashiontext.getString("REGISTER_TITLE")%></title>

<link rel=stylesheet href="<%=storeDir%>/fashionfair.css" type="text/css">
</head>

<body marginheight="0" marginwidth="0">

<!-- Set the user id and e-mail to the same value -->
<SCRIPT language="javascript">
function prepareSubmit(form)
{
        form.<%=ECUserConstants.EC_ADDR_EMAIL1%>.value =
                form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()
        form.<%=ECUserConstants.EC_UREG_LOGONID%>.value =
                form.<%= ECUserConstants.EC_UREG_LOGONID%>.value.toLowerCase()

        form.submit()
}
</SCRIPT>

<%
String incfile;

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


<%
incfile = "/" + storeDir + "/sidebar.jsp";
%>

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


<!--END SEARCH-->

<td bgcolor="#FFFFFF" width="600" rowspan="6" valign="top">


<!--MAIN CONTENT STARTS HERE-->

<table cellpadding="2" cellspacing="0" width="580" border="0" align="left">
<tr>

<td width="10" rowspan="10"> </td>


<td align="left" valign="top" colspan="3" class="categoryspace">
<font class="category"><%=infashiontext.getString("REGISTRATION")%></font>
<hr width="580" noshade align="left">
<font class="required">*</font><font class="text"><%=infashiontext.getString("REQUIRED_FIELDS3")%></font></td>
</tr><tr>
```

```
<td align="left" valign="top" width="400" class="topspace">
<%
if (strErrorMessage != null)
{
        //We have error message.
%>
<p><font color="red"><%=strErrorMessage%></font><br><br></p>
<%
}
%>
<FORM name="Register" method=POST action="<%="UserRegistrationAdd"%>">
<INPUT TYPE="hidden" NAME="langId" Value="<%=languageId%>">
<INPUT TYPE="hidden" NAME="new" Value="Y">
<INPUT TYPE="hidden" NAME="storeId" Value="<%=storeId%>">
<INPUT TYPE="hidden" NAME="catalogId" Value="<%=catalogId%>">
<INPUT TYPE="hidden" NAME="URL" Value="LogonForm">
<INPUT TYPE="hidden" NAME="page" Value="account">
<INPUT TYPE="hidden" NAME="registerType" Value="G">
<INPUT TYPE="hidden" NAME="profileType" Value="C">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_EMAIL1%>" Value="">
<!—
Lots of mandetory fields are not displayed in this form.
We set them to "-".
—>
<INPUT TYPE="hidden" NAME="personTitle" Value="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_ADDRESS1%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_ADDRESS2%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_CITY%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_STATE%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_ZIPCODE%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_COUNTRY%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%=ECUserConstants.EC_ADDR_PHONE1%>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%= ECUserConstants.EC_UREG_CHALLENGEQUESTION %>" VALUE="-">
<INPUT TYPE="hidden" NAME="<%= ECUserConstants.EC_UREG_CHALLENGEANSWER %>" VALUE="-">

<table cellpadding="3" cellspacing="0" border="0" align="left">
<tr>
<td align="right" valign="middle">
<font class="required">*</font><font class="strongtext"><%=infashiontext.getString("EMAIL2")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%= ECUserConstants.EC_UREG_LOGONID%>" value="<%=strLogonID%>"></td>
</tr><tr>
<td align="right" valign="middle">
<font class="required">*</font><font class="strongtext"><%=infashiontext.getString("PASSWORD3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%=ECUserConstants.EC_UREG_LOGONPASSWORD%>"
type="password" value="<%=strPassword%>"></td>
</tr><tr>
<td align="right" valign="middle">
<font class="required">*</font><font class="strongtext"><%=infashiontext.getString("VERIFY_PASSWORD3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%=ECUserConstants.EC_UREG_LOGONPASSWORDVERIFY%>"
type="password" value="<%=strPasswordVerify%>"></td>
</tr>
<% if (locale.toString().equals("ja_JP")||locale.toString().equals("ko_KR")||
locale.toString().equals("zh_CN")||locale.toString().equals("zh_TW")) { %>
```

```
<tr>
<td align="right" valign="middle">
<font class="required">*</font><font class="strongtext"><%=infashiontext.getString("LAST_NAME3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%=ECUserConstants.EC_ADDR_LASTNAME%>" value="<%=strLastName%>" type="text"></td>
</tr><tr>
<td align="right" valign="middle">
<font class="strongtext"><%=infashiontext.getString("FIRST_NAME3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%= ECUserConstants.EC_ADDR_FIRSTNAME %>" value="<%=strFirstName%>" type="text"></td>
</tr>
<% } else { %>
<font class="product">
<tr>
<td align="right" valign="middle">
<font class="strongtext"><%=infashiontext.getString("FIRST_NAME3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%= ECUserConstants.EC_ADDR_FIRSTNAME %>" value="<%=strFirstName%>" type="text"></td>
</tr><tr>
<td align="right" valign="middle">
<font class="required">*</font><font class="strongtext"><%=infashiontext.getString("LAST_NAME3")%></font></td>
<td align="left" valign="middle">
<input size="25" maxlength="50" name="<%=ECUserConstants.EC_ADDR_LASTNAME%>" value="<%=strLastName%>" type="text"></td>
</tr>
<% } %>
<tr>
<td align="left" valign="top">
<font class="text"> </font></td>
<td align="left" valign="top" class="categoryspace">
<table cellpadding="4" cellspacing="0" border="0">
<tr>
<td align="left" valign="middle" bgcolor="#FFCC99">
<A href="javascript:prepareSubmit(document.Register)"><font class="strongtext"><%=infashiontext.getString("SUBMIT")%></font></a>
</td>
</tr></table>
</td></tr></table>
</form>
</td>

<td width="180" valign="top" class="topspace">
<table cellpadding="3" cellspacing="0" border="1" width="180" bgcolor="#FFFFCC">
<tr>
<td align="left" valign="top">
<font class="strongtext"><%=infashiontext.getString("YOUR_PRIVACY")%></font><br>
<font class="text"><%=infashiontext.getString("PRIVACY_STATEMENT")%><br>
<a href="PrivacyView?langId=<%=languageId%>&storeId=<%=storeId%>&catalogId=<%=catalogId%>">
<%=infashiontext.getString("LEARN_MORE")%></a></font>
</td></tr></table>
</td>

</tr></table>
</td>

</tr></table>
<%
incfile = "/" + storeDir + "/footer.jsp";
%>

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

</body>

</html>
```

account.jsp

```
<%
//*****************************************************************
//*———————————————————————————————————————————————————————————————
//* Licensed Materials - Property of IBM
//*
//* 5697-D24
//*
//* (c) Copyright IBM Corp. 2000, 2002
//*
//* US Government Users Restricted Rights - Use, duplication or
//* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
//*
//*———————————————————————————————————————————————————————————————
//*
%>

<%@ page language="java" %>
<% // All JSPs requires the first 4 packages for getResource.jsp which is used for multi language support %>
<%@ page import="java.io.*" %>
<%@ page import="java.util.*" %>
<%@ page import="com.ibm.commerce.server.*" %>
<%@ page import="com.ibm.commerce.command.*" %>

<%@ page import="javax.servlet.*" %>
<%@ page import="com.ibm.commerce.catalog.beans.*" %>
<%@ page import="com.ibm.commerce.catalog.objects.*" %>
<%@ page import="com.ibm.commerce.beans.*" %>
<%@ page import="com.ibm.commerce.user.beans.*" %>
<%@ page import="com.ibm.commerce.user.objects.*" %>
<%@ page import="com.ibm.commerce.datatype.*" %>
<%@ page import="com.ibm.commerce.usermanagement.commands.ECUserConstants" %>
<%@ page import="com.ibm.commerce.security.commands.ECSecurityConstants" %>
<%@ page import="com.ibm.commerce.common.beans.*" %>
<%@ include file="getResource.jsp"%>

<%
CommandContext commandContext = (CommandContext)
        request.getAttribute(ECConstants.EC_COMMANDCONTEXT);

String catalogId = request.getParameter("catalogId");
String storeId = request.getParameter("storeId");
String languageId = request.getParameter("langId");

//Parameters may be encrypted.
if (catalogId == null)
        catalogId = ((String[]) request.getAttribute("catalogId"))[0];
if (storeId == null)
        storeId = ((String[]) request.getAttribute("storeId"))[0];
if (languageId == null)
        languageId = ((String[]) request.getAttribute("langId"))[0];
%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">

<head><title><%=infashiontext.getString("ACCOUNT_TITLE")%></title>
<link rel=stylesheet href="<%=storeDir%>/fashionfair.css" type="text/css">
</head>

<body marginheight="0" marginwidth="0">

<%
String incfile;

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


<%
incfile = "/" + storeDir + "/sidebar.jsp";
%>

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

<!—END SEARCH—>

<td bgcolor="#FFFFFF" width="600" rowspan="6" valign="top">

<%
//Deal with possible errors when logging in
String strPageTitle      = "Logon";
String strErrorMessage   = null;
String strErrorCode      = null;

String[] strArrayAuth = (String [])request.getAttribute(ECConstants.EC_ERROR_CODE);

if (strArrayAuth != null){
     if(strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_DISABLED_ACCOUNT) == true){
        strErrorMessage = infashiontext.getString("ACCOUNT_LOCKED");
     }else if( strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_MISSING_LOGONID) == true){
        strErrorMessage = infashiontext.getString("LOGIN_ID_MISSING");
     }else if(strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_INVALID_LOGONID) == true){
        strErrorMessage = infashiontext.getString("LOGON_ID_INVALID");;
     }else if(strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_MISSING_PASSWORD) == true){
        strErrorMessage = infashiontext.getString("PASSWD_MISSING");
     }else if(strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_INVALID_PASSWORD) == true){
        strErrorMessage = infashiontext.getString("PASSWORD_INCORRECT");
     }else if(strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_LOGON_NOT_ALLOWED) == true){
        strErrorMessage = infashiontext.getString("WAIT_TO_LOGIN");

     }
}
```

```
/*
 * Register link behaves differently depending on if the user is logged
 * in. If the user is logged in (hence a registered user) clicking on
 * register will log the user off then display the registration form.
 * This will allow multiple registration using the same browser.
 * For guest shoppers clicking on Register link will simply display the
 * registration form.
 */
UserRegistrationDataBean regBean = new UserRegistrationDataBean();
com.ibm.commerce.beans.DataBeanManager.activate(regBean, request);
String regURL = null;

if (! regBean.findUser())
// findUser() return false if the customer is not registered
{
        //This is a guest user. Simply display the registration form.
        regURL = ″UserRegistrationForm″;
}
else
{
        //This is a registered/logged in user. Log him off first.
        //Through LogoffView registration page will be displayed.
        regURL = ″Logoff″;
}
%>
<!—MAIN CONTENT STARTS HERE—>

<table cellpadding=″2″ cellspacing=″0″ width=″580″ border=″0″>
<tr>

<td width=″10″ rowspan=″10″> </td>


<td align=″left″ valign=″top″ colspan=″3″ class=″categoryspace″>
<font class=″category″><%=infashiontext.getString(″MY_ACCOUNT3″)%></font>
<hr width=″580″ color=″#336666″ noshade align=″left″>
</td>

</tr><tr>



<td align=″left″ valign=″top″ width=″280″ bgcolor=″#CC6600″>
<font class=″subheader″><%=infashiontext.getString(″PERSONAL_INFO″)%></font></td>

<td width=″20″ rowspan=″5″>  </td>

<td align=″left″ valign=″top″ width=″280″ bgcolor=″#CC6600″>
<font class=″subheader″><%=infashiontext.getString(″ADDRESS_BOOK″)%></td>

</tr><tr>

<td align=″left″ valign=″top″ width=″280″ class=″topspace″>
<font class=″text″><%=infashiontext.getString(″UPDATE_NAME″)%></font><p>

<table cellpadding=″0″ cellspacing=″0″ border=″0″ align=″left″>
<tr>
<td align=″left″ valign=″top″>
```

```
<table cellpadding="4" cellspacing="0" border="0">
<tr>
<td align="left" valign="middle" bgcolor="#FFCC99">
<A href="UserRegistrationForm?storeId=<%=storeId%>&langId=<%=languageId%>&catalogId=<%=catalogId%>">
<font class="strongtext"><%=infashiontext.getString("CHANGE_INFO")%></font></a></td>
</tr></table>


</td>
</tr></table>


</td>

<td align="left" valign="top" width="280" class="topspace">

<%
if (strErrorMessage != null)
{
        //We have an error message.
%>
<p><font color="red" ><%=strErrorMessage%></font><br>
<%
}
%>
<table cellpadding="0" cellspacing="0" border="0">
<tr>
<td align="left" valign="top">
<font class="text"><%=infashiontext.getString("UPDATE_ADDRESS1")%></font><br><br></td>
</tr><tr>

<td align="left">
<table cellpadding="4" cellspacing="0" border="0">
<tr>
<td align="left" valign="middle" bgcolor="#FFCC99">
<A href="AddressBookForm?storeId=<%=storeId%>&langId=<%=languageId%>&catalogId=<%=catalogId%>">
<font class="strongtext"><%=infashiontext.getString("EDIT_ADD")%></font></a></td>
</tr></table>
</td></tr></table>


</tr></table>
</td>

</td>
</tr></table>
<%
incfile = "/" + storeDir + "/footer.jsp";
%>

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

</body>

</html>
```

infashiontext_en_US.properties

LOGON_ID_INVALID = The e-mail address entered is invalid. Type another e-mail address in the E-mail address field and try again.
PASSWORD_INCORRECT = The password entered is incorrect. Type your password in the Password field and try again.
LOGIN_ID_MISSING = Type an e-mail address in the E-mail address field.
PASSWD_MISSING = Type a password in the Password field.
ACCOUNT_LOCKED = Due to 3 unsuccessful password attempts, you will be unable to logon. Please contact a store representative to unlock your account.
WAIT_TO_LOGIN = Please wait a few seconds before attempting to log in again.

PASS_ERROR_MESSAGE21 = You entered a password with less than 6 characters. Passwords must be at least 6 characters in length, and include one digit and one letter. Please re-enter your password.
PASS_ERROR_MESSAGE22 = Your password does not contain a digit. Passwords must be at least 6 characters in length, and include one digit and one letter. Please re-enter your password.
PASS_ERROR_MESSAGE23 = Your password does not contain a letter. Passwords must be at least 6 characters in length, and include one digit and one letter. Please re-enter your password.
PASS_ERROR_MESSAGE24 = Your password is the same as your user-id. Please assure that your user-id and password are different.
PASS_ERROR_MESSAGE25 = Your new password is the same as the previous one. Please enter a new password, or choose 'My Account' on the menu bar to return to your account page.
PASS_ERROR_MESSAGE26 = A character in your password occurs more consecutively than the allowed limit of 3. Please re-enter your password.
PASS_ERROR_MESSAGE27 = A character in your password occurs more than the allowed limit of 4. Please re-enter your password.

infashion store translation text

ENCODESTATEMENT = text/html; charset=ISO_8859-1

# header.jsp
SHOPPING_CART = SHOPPING CART
MY_ACCOUNT = MY ACCOUNT
CONTACT_US = CONTACT US
HELP = HELP
SEARCH = SEARCH
HOME = Home

#footer.jsp
SHOPPING_CART2 = Shopping cart
MY_ACCOUNT2 = My account
CONTACT_US2 = Contact us
HELP2 = Help
PRIVACY_POLICY = Privacy policy

#sidebar.jsp
CHOOSE_COUNTRY = CHOOSE A LANGUAGE
CANADA = Canada
UNITED_STATES = United States
CHINA = China
FRANCE = France
GERMANY = Germany
ITALY = Italy
JAPAN = Japan
SPAIN = Spain
TURKEY = Turkey
SERVICES = SERVICES
REGISTER = Register
NOW_BUY = now and get 10% off your first purchase!
NEED_HELP = Need help?
JUST_ASK = Just ask!
GO = GO

.
.
.

Appendix H. Troubleshooting

This section lists potential problems you may encounter during migration and actions to resolve these problems.

+ • **Problem**: You receive a 404 error in your browser when you attempt to access
+ WebSphere Commerce tools such as the WebSphere Commerce Administration
+ tool.

+ **Solution**: If the WebSphere Commerce application server seen from the
+ WebSphere Application Server Administrative Console is using a virtual host
+ other than the `default_host` (VH_*instancename*) do the following:

+ 1. Delete this virtual host.
+ 2. Configure the WCS Stores module, which can be found under WebSphere
+ Commerce Enterprise Application - *instance_name* in the Administration
+ Console to use the `default_host` virtual host instead of the other virtual host.
+ 3. Restart the WebSphere Commerce application server.
+ 4. Regenerate the Webserver plug-in.

+ For more information on how to regenerate the Web server plug-in, refer to
+ the section, "Regenerate plug-in" on page 337 of the Redbook *WebSphere*
+ *Commerce V5.4 Handbook Architecture and Integration Guide* (sg246567.pdf).

+ You should leave the WCS Tools module to use the VH_ *instancename*_tools
+ as it is.

• **Problem**: The Store Archive page in the Store Services utility indicates that your store which you published in Commerce Suite 5.1 and have now migrated to WebSphere Commerce 5.4 has a status of `not published`.

Solution: When migrating to WebSphere Commerce 5.4, the database migration script clears all the entries which have SCSSTATE set to 'C' (complete) in the SCHSTATUS table. The database migration script moves all other entries in the SCHSTATUS table that are in a pending or incomplete state to the SCHACTIVE table. As the entries that were previously set to complete are not moved to the new SCHACTIVE table, your store's state will display as `not published` in Store Services after migration.

Note that your store will continue to function properly.

• **Problem**: When you are running the database migration script for DB2, you receive an error message similar to the following (usually in the generated `message.txt` file):

```
001.10.01 14:20:19.027 java.lang.Class executeUpdate   MassLoader
An SQL Exception was received [IBM][CLI Driver][DB2/NT]
SQL0964C  The transaction log for the database is full.
SQLSTATE=57011  on record [attribute: [ATTRIBUTE_ID=10176][LANGUAGE_ID=-3]
  [ATTRTYPE_ID=STRING][NAME=Farbe][SEQUENCE=1.0][DESCRIPTION=Farbe]
  [CATENTRY_ID=11040][DESCRIPTION2=Farbe]].
[sqlcode: -964, sqlstate: 57011]
```

Solution: Basically, the above message indicates that the DB2 database log size is too small for the operation being performed. The following database configuration parameters control the log size:

– LOGFILSIZ
– LOGPRIMARY
– LOGSECOND

As an example, if LOGFILSIZ is 1000 (4KB), and LOGPRIMARY is 40, then you'll have a total log size of : 1000 * 4K * 40 = 160 MB. If the total number of uncommitted transactions require more than this amount of space, then LOGSECOND gets used. If after that, there is still not enough space, you receive the above error message. If you are developing your application, then you should do commits more often in your application.

To display the current parameters for your database enter the following statement from a DB2 command window:

```
DB2 GET DB CFG FOR db_name
```

where *db_name* is the name of your database, for example, MALL.

To increase the log size, enter the following statement from a DB2 command window:

```
DB2 UPDATE DB CFG FOR db_name USING LOGFILSIZ nnn
```

where *db_name* is the name of your database, for example, MALL, and *nnn* is a number representing the LOGFILSIZ value.

See the DB2 administration information for more details on setting up these configuration parameters.In addition, you can drop the following summary tables if they exist on your system as these summary tables can contribute to the transaction log overflow. Use the following SQL statement to drop your tables:

```
DB2 DROP TABLE table_name
```

where *table_name* is one of the following summary tables:

– PRODUCT
– STOREINV
– CATEGORY
– RICHATTR
– RICHATTRCATGP

Note: The PRODUCT table is the most likely to cause the transaction log to overflow, so delete it first and try to publish the store before deleting the other tables.

You can also create secondary transaction logs using the DB2 Control Center. For more information, see the *DB2 Administration Guide*.

- **Problem**: When attempting to register a new user in the migrated WebFashion store, it fails with an error similar to:

```
...
TimeStamp:    2001-10-13 18:17:46.456
Thread ID:    <Worker#2>
Class:        com.ibm.commerce.infashion.commands.RegisterNAddToMemberGroupImpl
Method:       performExecute
Severity:     1
Message Text: CMN0411E The following Finder Exception occurred during processing:
              "javax.ejb.ObjectNotFoundException".
Exception:    javax.ejb.ObjectNotFoundException
        at javax.ejb.FinderException.<init>(FinderException.java:36)
        at javax.ejb.ObjectNotFoundException.<init>(ObjectNotFoundException.java:38)
        ...
```

Solution: Update the register.jsp by replacing the command RegisterNAddToMemberGroup with the command UserRegisterAdd, and change the customer profile as described in "User registration for sample WebFashion store" on page 113

- **Problem**: **If you are using Oracle**, and you see errors like the following when you view the data migration log file:

```
ERROR at line 1:   ORA-00997: illegal use of LONG datatype
```

  Solution: The problem is that you cannot use the LONG as a typical data type. Look at the Oracle documentation for any resolutions. It is suggested that if you receive the above message, you take the following steps on your Commerce Suite 5.1 data, *before any migration takes place*. If you have begun the migration process, you will likely need to load a back up copy of your Commerce Suite 5.1 data:

  1. Export the table with the columns defined as LONG to a file (for example, table CATENTDESC).
  2. Drop or delete the table.
  3. Import the table using the indexfile option. This will create a file that will create the table.
  4. Edit the file created in the above step, and change the column definitions to the one that you want. Refer to the indexfile.sql sample file provided in the /usr/WebSphere/CommerceServer/schema/oracle/migration directory.
  5. Run the file you edited in the above step to create the new table.
  6. Re-import the data by running import with ignore=y.
  7. Start the data migration process once again.

  In cases where the migration script cannot simply add or modify columns for a table, it is necessary to create a temporary table for the Commerce Suite 5.1 table during the data migration and then create the new WebSphere Commerce 5.4 table. The migration script moves the data from the temporary Commerce Suite 5.1 table to the new WebSphere Commerce 5.4 table using the following SQL statement:

```
insert into 52_table (list of comma separated 52 columns)
    select list of comma separated 51 columns  from 51_temporary_table
```

  This type of SQL statement fails if there are any Commerce Suite 5.1 columns that are defined as LONG.

- **Problem**: If you have customized tables that contain foreign key links to the shipped Commerce Suite 5.1 tables, the data migration script may fail when it attempts to drop the referential integrity constraints (such as foreign keys, primary keys, and indexes) during data migration. The following is a typical error message:

```
ERROR at line 1:
ORA-02273: this unique/primary key is referenced by some foreign keys
```

  Solution: You need to drop any of the constraints that are connected to Commerce Suite 5.1 tables before migration, and then restore the constraints after migration to WebSphere Commerce 5.4. If you have begun the migration process, you will likely need to load a back up copy of your Commerce Suite 5.1 data.

If you have modified any of the Commerce Suite 5.1 referential integrity constraints, that is, you have added an extra column to an already existing index or foreign key relationship, the column will be deleted as part of the data migration process.

- **Problem**: After migrating to WebSphere Commerce 5.4, when you log on to the WebSphere Commerce 5.4 Administration Console and try to modify or create a user (that is, select a user from the list and click **Change**) the console displays a blank page. The following are the errors in the ecmsg.log file:

```
==============
TimeStamp:     2002-02-04 19:04:40.908
Thread ID:     <Servlet.Engine.Transports:10>
Class:         UIPropertiesBean
Method:        setRequestProperties
Severity:      1
Message Text: CMN7023E The system could not pass the XMLFile parameter into
               the UI Property Element.  The element could not be loaded.


==============
TimeStamp:     2002-02-04 19:04:41.036
Thread ID:     <Servlet.Engine.Transports:10>
Class:         HttpForwardViewCommandImpl
Method:        performExecute
Severity:      1
Message Text: CMN1244E An error occurred when forwarding document
               tools/common/NotebookNavigation.jsp.javax.servlet.ServletException
Exception:     javax.servlet.ServletException at
               org.apache.jasper.runtime.PageContextImpl.
               handlePageException(PageContextImpl.java:604)
               at tools.common._NotebookNavigation_jsp_0._jspService
               (_NotebookNavigation_jsp_0.java:314)
               at org.apache.jasper.runtime.HttpJspBase.
               service(HttpJspBase.java:142)
     .
     .
     .
```

Solution: This exception arises as the tools are finding a previous version of the UserRegistration_*locale*.properties file. This will also cause a problem in the CSA tool for displaying customer profiles, and in stores while registering a new user. For WebSphere Commerce 5.4, these files should only reside in the /usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear/properties directory. To fix this problem, you need to find all instances of WebSphere Commerce Suite 5.1 UserRegistration_*locale*.properties files in your system that are not in the above directory, and either rename the folder they are in, or delete the files. After renaming the folder or deleting the files, restart WebSphere Application Server. You should now be able to change a user in the WebSphere Commerce 5.4 Administrator.

- **Problem**: After migrating to WebSphere Commerce 5.4, you may encounter an unexpected access control violation which has an error message similar to the following in the message log file (by default , it is set to ecmsg_xxxx.log):

```
TimeStamp:     2002-02-15 09:03:24.14
Thread ID:     <Servlet.Engine.Transports:10>
Class:         AccManager
Method:        isAllowed
Severity:      1
Message Text: CMN1501E User 859 does not have the authority to perform action
"com.ibm.commerce.usermanagement.commands.OrgEntityAddCmd" on resource
"com.ibm.commerce.user.objects._Organization_Stub" for command "OrgEntityAdd".
```

Solution: This exception means that the resource level access control check failed. It is resource level access control failure as opposed to command level access control failure, because the action is a WebSphere Commerce command. If it was a command level access control failure, the action would be Execute. In this case, the action is the OrgEntityAdd command.

In WebSphere Commerce Suite 5.1, resource level access control was enforced programmatically, within the command logic. In WebSphere Commerce 5.4, resource level access control policies are specified externally, similar to the way command level access control policies are specified. During the migration, only command level access control policies are migrated from Commerce Suite 5.1 to WebSphere Commerce 5.4. Any resource-level access control policies that are needed due to customization of the Commerce Suite 5.1 default access control policies (which are stored in the ACCCMDGRP table), need to be added manually, as demonstrated below.

Based on the WebSphere Commerce 5.4 default access control policies, only membership administrators and site administrators have access to the OrgEntityAdd command. In this example, if you want user 859 (shown in the above log and who does not have any special role) to access this command, you need to either change the existing policy or add a new one. This policy should refer to a less restrictive access group (that is, a grouping of users for access control purposes) that would include user 859. An example is the AllUsers access group, which includes all users. The new access control policy would also have to include the com.ibm.commerce.usermanagement.commands.OrgEntityAddCmd action, as determined from the log, in its action group. The resource group would have to include the com.ibm.commerce.user.objects.Organization resource, as determined from the log. Note that this is the remote interface of the Organization EJB. Also note that this is slightly different from the name of the resource shown in the log. The reason for this difference is that the command returns the Organization access bean when it specifies its resources to protect, however, the WebSphere Commerce runtime then has to narrow this access bean to its remote interface since it is the remote interface that extends the Protectable interface. See the *WebSphere Commerce 5.4 Programmer's Guide* for more details about protecting EJB resources.

Most of the necessary components for this sample policy (AccessGroup, ActionGroup, ResourceGroup) have already been defined in the default access control policies; this simplifies the process for adding a new access control policy. Here is the existing resource level policy, as specified in the default access control policies file, defaultAccessControlPolicies.xml:

```
<Policy Name="MembershipAdministratorsForOrgExecuteOrgEntityRegistrationCommandsOnOrganizationResource"
    OwnerID="RootOrganization"
    UserGroup="MembershipAdministratorsForOrg"
    ActionGroupName="OrgEntityRegistration"
    ResourceGroupName="OrganizationDataResourceGroup"
    PolicyType="template"/>
</Policy>
```

Note that the UserGroup specifies MembershipAdministratorsForOrg. In this example, the goal is to add a new policy that includes all users.

Create the new, less restrictive resource level access control policy in a file, for example orgentityaddpolicy.xml. This file must be placed in the /usr/WebSphere/CommerceServer/xml/policies/xml directory. Once it has been defined, you can load it by running the following script found in the

/usr/WebSphere/CommerceServer/bin directory. Run the script as follows. (See the *WebSphere Commerce 5.4 Access Control Guide* for more details on running this script.)

```
acpload database userid password orgentityaddpolicy.xml
```

Here are the contents of `orgentityaddpolicy.xml`:

```
<?xml version="1.0"encoding="ISO-8859-1"standalone="no"?>

<!DOCTYPE Policies SYSTEM "../dtd/accesscontrolpolicies.dtd">
<Policies>

<Policy Name="AllUsersExecuteOrgEntityRegistrationOnOrganizationResourceGroup"
OwnerID="RootOrganization"
UserGroup="AllUsers"
ActionGroupName="OrgEntityRegistration"
ResourceGroupName="OrganizationDataResourceGroup">

</Policy>

<Policies>
```

- **Problem**: After migrating to WebSphere Commerce 5.4, you see the following error when you bring up any JavaServer Pages, or click on some selection on the JSP after migration:

```
TimeStamp:    2002-02-27 11:24:17.04
Thread ID:    <Servlet.Engine.Transports:10>
Class:        HttpForwardViewCommandImpl
Method:       performExecute
Severity:     1
Message Text: CMN1244E An error occurred when forwarding document
              llbean/productvertical.jsp.org.apache.jasper.JasperException:
              /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/
                  wcstores.war/xxxx/productvertical.jsp(2676,0)
              JSPG0059E: Unable to compile class for JSP
              null/usr/WebSphere/AppServer/temp/prizm/WebSphere_Commerce_Server_-_demo/
                  WebSphere_Commerce_Enterprise_Application_-_demo/wcstores.war/
                  xxxx/_productvertical_jsp_3.java:57:
            Class com.ibm.util.Sorter not found in import.
              import com.ibm.util.Sorter;
```

Solution: Your customized code and JSP has been using the `com.ibm.util.Sorter` class from IBM WebSphere Application Server 3.5.x. The entire `com.ibm.util` package has been removed from the current version of WebSphere Application Server. You need to rewrite your code and JSP by using the equivalent class from the IBM SDK for Java packaged with WebSphere Application Server 4.0.2.

- **Problem**: **If you are using Oracle**, and have migrated to WebSphere Commerce 5.4 with the ATP option selected when you ran the database migration script, and you find the following errors in the `migratedb.log`:

```
              ALTER TABLE rcptavail...
ADD  ( CONSTRAINT f_659                          *
ERROR at line 2:
    ORA-02298: cannot validate (WCSADMIN.F_659) - parent keys not found
```

Or, if you did not notice the above error in `migratedb.log`, and when you attempt to publish a new store, it fails with the following errors in the `messages.txt` file in the `/usr/WebSphere/CommerceServer/logs/migration` directory:

```
2002.02.27 13:32:20.763 java.lang.Class formatValueToObject   MassLoader
Error when formatting value for RCPTAVAIL.RECEIPT_ID :
  @receipt_id_105 with error [class java.lang.NumberFormatException(@receipt_id_105)].
```

Solution: Execute the following query on your database:

```
select constraint_name,constraint_type from all_constraints
  where owner='WC_SADMIN' and table_name='RCPTAVAIL;'
```

You should see output similar to:

```
CONSTRAINT_NAME              C
---------------------------- -
F_658                        R
SYS_C00147031                C
SYS_C00147032                C
SYS_C00147033                C
SYS_C00147875                P
```

If the constraint name F_659 is missing, you can manually fix it as follows:

1. Check the records in table RCPTAVAIL. Delete those records that do not have RECEIPT_ID in the table RECEIPT.

   – To display the records in table RCPTAVAIL that do not have RECEIPT_ID in table RECEIPT run the following SQL statement:

   ```
   SELECT receipt_id FROM rcptavail WHERE receipt_id
      NOT IN (SELECT receipt_id FROM receipt);
   ```

   – To delete records as displayed from the above select statement:

   ```
   DELETE FROM rcptavail WHERE receipt_id
      NOT IN (SELECT receipt_id FROM receipt);
   ```

2. Recreate the foreign key on table RCPTAVAIL to table RECEIPT using the following SQL statement:

   ```
   ALTER TABLE rcptavail
     ADD CONSTRAINT f_659
     FOREIGN KEY (receipt_id)
     REFERENCES receipt
     ON DELETE CASCADE;
   ```

3. Try to publish the new store successfully.

- **Problem**: After registering a new user ID, you cannot logon with the user ID.

  Solution: The logon may be failing because your custom Logon command is calling individual task commands directly instead of simply wrapping the out-of-box Logon command. For example, it may be because your custom Logon command is calling com.ibm.commerce.security.commands.AuthenticationPolicyCmdImpl which should only be called during registration in WebSphere Commerce 5.4.

  To bypass this problem, you can modify *your_instance*.xml from:

  ```
  <Security
  ...
  passwordpolicy="true"/>
  ```

  to

  ```
  <Security
  ...
  passwordpolicy="false"/>
  ```

  This will bypass the login inside the AuthenticationPolicyCmdImpl class

- **Problem**: After deploying the /usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear file to WebSphere Application Server and starting the WebSphere Commerce server, you encounter problems with JNDI.

  Solution: Check your JNDI name. Usually after importing the ear file, the JNDI name may be changed. You will need to modify the JNDI name manually from the WebSphere Application Server Administrative Console. You should change the JNDI names of all your EJB groups according to the JNDI name in your

source files. For example, change from demoOrderItemExt to democom/ibm/wcsx/some_company/order/objects/OrderItemExt.

- **Problem**: After importing the /usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear file to WebSphere Application Server using XMLConfig, and starting the server, WebSphere Commerce Server – *instance_name*, WebSphere Application Server reports a servlet initialization error from the Administrative Console similar to:

```
Did not realize init() exception thrown by servlet WCS Stores Request Servlet:
javax.servlet.ServletException:
*** Error: Initialization passing property parameters to EJB failure.
```

  Solution: This may indicate that the /usr/WebSphere/AppServer/installedApps/ WC_Enterprise_App_*instance_name*.ear file has not been installed correctly using XMLConfig due to problems with your environment, even though XMLConfig reports no error when importing the .ear file. You can remove the Enterprise Application (WebSphere Commerce Enterprise Application - *instance_name*) from the Administrative Console, and then redeploy the .ear file again. Usually by doing this, the WebSphere Commerce Server – *instance_name* server can be started correctly.

- **Problem**: When running the instance migration script, you receive a NullPoint Exception.

  Solution: The <migrationFrom> element in the product.xml file was not updated as described in "Updating product.xml" on page 66; this step is required if the previous version of WebSphere Commerce is not on the same machine as WebSphere Commerce 5.4.

- **Problem**: When running the instance migration script as wasuser, you receive the following error:

```
java.io.FileNotFoundException:
/usr/WebSphere/CommerceServer/instances/instance_name/logsdemo_export.xml
(A file or directory in the path name does not exist.)
```

  Solution: This may be due to the existence of another instance of an *instance*.xml file (for example, *instance*.xml.bak) in the /usr/WebSphere/CommerceServer/instances/*your_instance*/xml directory. To fix this problem, you can just remove the additional instance of the file from this directory before running the instance migration script.

- **Problem**: **If you are using Oracle** and you complete your migration without any problem, but when you display your "Publish Summary" page you receive exceptions similar to the following:

```
[3/22/02 13:17:35:875 CST] 5ae55e04 HttpTransport A SRVE0171I: Transport http is listening on port 9,081.
[3/22/02 13:17:35:890 CST] 5ae55e04 Server      A WSVR0023I: Server WebSphere Commerce Server - demo
  open for e-business
[3/22/02 13:18:52:937 CST]  6cb9e51 ExceptionUtil X CNTR0019E: Non-application exception occurred while processing
  method findByPrimaryKey: com.ibm.websphere.cpi.CPIException:
  com.ibm.ejs.cm.portability.TableDoesNotExistException: ORA-00942: table or view does not exist;
  nested exception is:
 com.ibm.ejs.cm.portability.TableDoesNotExistException: ORA-00942: table or view does not exist

.
.
.
```

  Solution: Change the line compatible = 8.0.5 to compatible = 8.1.0 in the init*wcs_database_name*.ora file.

- **Problem**: You are experiencing problems with the co-existence of a newly published WebSphere Commerce 5.4 store with a migrated version of that store from WebSphere Commerce Suite 5.1. For example, if you have published the WebSphere Commerce Suite 5.1 InFashion store in WebSphere Commerce Suite 5.1, and then you migrate that store to WebSphere Commerce 5.4 and re-publish it. After migrating to WebSphere Commerce 5.4, you also publish the WebSphere

Commerce 5.4 InFashion store. When you launch the migrated WebSphere Commerce Suite 5.1 InFashion store, or the newly published WebSphere Commerce 5.4 InFashion store, there would be more products in the CatalogDisplay page. If you click on the new displayed product, you receive the exceptions similar to:

```
==============
TimeStamp:     2002-04-12 18:06:42.912
Thread ID:     <Servlet.Engine.Transports:9>
Class:         com.ibm.commerce.catalog.commands.ProductDisplayCmdImpl
Method:        performExecute
Severity:      1
Message Text: CMN1222E Either the Catalog Entry "10,849" does not exist in the catalog,
               or it cannot be processed in store "10,001".
Exception:     javax.ejb.ObjectNotFoundException
               at com.ibm.commerce.catalog.objects.EJSJDBCPersisterCMPCatalogEntryBean.
               findByKeyAndStore(EJSJDBCPersisterCMPCatalogEntryBean.java:692)


.
.
.
```

Solution: You can solve this problem in either of the following ways:

– Manually modify the `catalog.xml` file packaged in the WebSphere Commerce 5.4 InFashion `.sar` file before publishing it. Change the `IDENTIFIER` column of the `CATALOG` table from `InFashions` to `InFashion`**54**. The ID resolver will generate a new catalog ID for the newly published WebSphere Commerce 5.4 InFashion store that will not conflict with the catalog ID of the migrated InFashion store.

– When you publish a WebSphere Commerce 5.4 InFashion `.sar` file, you should choose a new organization so that the catalog ID does not conflict with the migrated WebSphere Commerce Suite 5.1 InFashion store as a new catalog ID will be generated by ID Resolver that will not conflict with the catalog ID of the migrated InFashion store.

Enabling tracing information

When enabled, the trace information file, `ecmsg_xxxx.log` is generated in the `/usr/WebSphere/CommerceServer/instances/`*instance_name*`/logs` directory. To enable the generation of this file which logs the different traces for the WebSphere Commerce 5.4 server component, it is recommended that you configure the Log System node in Configuration Manager to enable tracing. Alternatively, you can add the following to the trace tag in the *instance_name*`.xml` file, however, exercise caution when manually editing this file.

```
<LogSystem name="Log System">
    <trace fileSize="40"
           display="false"
           traceFile="/usr/lpp/CommerceServer/instances/instance_name/logs/ecmsg.log" >
      <component name="SERVER" />
      <component name="CATALOG" />
      <component name="ORDER" />
      <component name="USER" />
      <component name="COMMAND" />
      <component name="PVC" />
      <component name="UBF" />
      <component name="INVENTORY" />
      <component name="RFQ" />
      <component name="REPORTING" />
      <component name="TOOLSFRAMEWORK" />
      <component name="DEVTOOLS" />
    </trace>

    <messageLog fileSize="40"
                messageFile="/usr/lpp/CommerceServer/instances/instance_name/logs/ecmsg.log"
```

```
              notification="false"
              display="false">
    <logSeverity type="ERROR" />
  </messageLog>
  <activityLog display="false">
    <userTraffic cacheSize="20" />
    <accessLogging logAllRequests="false"
                  cacheSize="32" />
  </activityLog>
</LogSystem>
```

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.

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

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

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

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

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

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

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

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

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

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.

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.

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

Trademarks

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

AIX	AIX 5L	DB2
DB2 Universal Database	IBM	Net.Commerce
RS/6000	SecureWay	Tivoli
WebSphere		

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

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

SET and the SET Logo are registered trademarks owned by SET Secure Electronic Transaction LLC in the United States, other countries, or both.

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

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

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

IBM®

Printed in USA