

Keeping Commerce Applications Updated

WebSphere Commerce 5.1 to 5.6 Migration Guide

Migration Strategy and Planning

Production and Development Environments

Step-by-Step Instructions



Hernan Cunico
Andrew Hays
Steve Insley
Khurram Wyne
Nicolai Nielsen
Sanjeev Sharma
Sanjay Shah
Drake Philbrook



International Technical Support Organization

**Keeping Commerce Applications Updated
WebSphere Commerce 5.1 to 5.6 Migration Guide**

July 2004

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (July 2004)

This edition applies to Version 5, Release 6, Modification 0 of WebSphere Commerce.

This document created or updated on July 28, 2004.

© Copyright International Business Machines Corporation 2004. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xi
Trademarks	xii
Preface	xiii
The team that wrote this redbook	xiii
Become a published author	xv
Comments welcome	xv
Part 1. Introduction to WebSphere Commerce V5.6	1
Chapter 1. Introduction	3
1.1 Overview	4
1.2 Structure of the book	4
Chapter 2. WebSphere Commerce V5.6 Overview	7
2.1 Product overview	8
2.2 WebSphere Commerce software components	10
2.2.1 Web server	11
2.2.2 WebSphere Application Server	11
2.2.3 Database Server	12
2.2.4 WebSphere Commerce Server	12
2.2.5 WebSphere Commerce Payments Server	13
2.2.6 Enablement software	13
2.3 WebSphere Commerce Server subsystems	15
2.3.1 Member subsystem	15
2.3.2 Catalog subsystem	19
2.3.3 Trading subsystem	20
2.3.4 Order subsystem	20
2.3.5 Merchandising subsystem	20
2.3.6 Marketing subsystem	20
2.3.7 Inventory subsystem	20
2.3.8 Messaging subsystem	21
2.4 WebSphere Commerce Tools	21
2.5 WebSphere Commerce Business models	22
2.5.1 Direct Sales	23
2.5.2 Value Chain	24
2.5.3 Hosting Ex-Sites	25
2.6 What's new in WebSphere Commerce V5.6	26
2.6.1 WebSphere Commerce Development Environment	26

2.6.2	WebSphere Commerce Payments	27
2.6.3	Configuration Manager	27
2.6.4	Loader Package	27
2.6.5	Password Manager	27
2.6.6	Adaptor for CrossWorld	28
2.6.7	Catalog and product management	28
2.6.8	Business Models	30
2.6.9	Access Control	30
2.6.10	Collaboration	30
2.6.11	Campaigns and Promotions	31
2.6.12	Order and inventory Management	31
2.6.13	Analytics	32
2.6.14	Security	32
2.6.15	Caching	33
2.6.16	Product Advisor	35
Chapter 3.	Migration Strategy and Planning	37
3.1	Migration Strategy considerations	38
3.1.1	Add or replace functionality	38
3.1.2	Take advantage of DynaCache	38
3.1.3	Optimization	38
3.1.4	External product and user management	39
3.1.5	How does custom code impact the process?	39
3.1.6	Prepare a detailed Plan	39
3.2	Migration Planning	40
3.2.1	Skill requirements	40
3.2.2	Hardware and software prerequisites	41
3.3	Product versions mapping	44
3.4	Migration approach for development environment	44
3.5	Migration approaches for runtime environment	45
3.5.1	Switch-Over migration scenario	46
3.5.2	Co-existence migration scenario	48
3.5.3	In-place migration scenario	48
3.6	Approach used for this migration	49
3.6.1	Detailed overview of development migration steps	50
3.6.2	Detailed overview of production environment migration	51
Chapter 4.	Commerce Application used during the migration	55
4.1	Commerce application	56
4.1.1	Application environment	56
4.2	Commerce customization	57
4.2.1	Shopping flow	57
4.2.2	Commands	60

4.2.3 Database tables	62
4.3 Catalog	62
4.4 Database	63
Part 2. Development environment	65
Chapter 5. Installing WebSphere Commerce Development Environment	67
5.1 Development environment overview	68
5.2 Pre-installation requirements	68
5.2.1 VisualAge for Java V4.0 prerequisites	69
5.3 Installing WebSphere Studio Application Developer	69
5.3.1 Pre-installation steps	69
5.3.2 Installing WebSphere Studio Application Developer V5.1.1	70
5.3.3 Apply fixes to the test environment	71
5.4 Installing WebSphere Commerce V5.6 Toolkit	75
5.4.1 Installing WebSphere Commerce V5.6 Toolkit	76
5.5 Installing DB2 Universal Database V8.1	77
5.5.1 Backup development databases	77
5.5.2 Installation requirements	78
5.5.3 Preparing a Windows user for DB2	79
5.5.4 Installing DB2 Universal Database V8.1	80
5.5.5 Installing DB2 Universal Database fixpack 5	83
5.5.6 Migrate databases to DB2 Universal Database V8.1 level	85
5.6 Configuring the development environment for DB2	85
5.7 Installing VisualAge for Java V4.0	87
5.7.1 Installing VisualAge for Java V4.0	87
5.7.2 Configuring VisualAge for Java V4.0	89
Chapter 6. Pre-migration steps	95
6.1 Preparation overview	96
6.2 Instance preparation	97
6.2.1 Updating the product information file	97
6.2.2 Prepare resources	99
6.2.3 Merge EJB groups	100
6.2.4 Export code from VisualAge for Java V3.5.3	100
6.3 Database preparation	102
6.3.1 Update database configuration values	102
6.3.2 Unsent messages	104
6.3.3 Order status	104
6.3.4 Catalog	104
6.3.5 Members	105
6.3.6 Custom message types	108
6.3.7 Dropping foreign key references	108
6.3.8 Erroneous data in encrypted fields	109

Chapter 7. Migrating the development environment	111
7.1 Current development environment	112
7.2 Development environment migrated	112
7.3 Migrating the development instance	113
7.3.1 Overview	113
7.3.2 Running the tool	114
7.4 Migrating the development database	116
7.4.1 Migrating unencrypted data	116
7.4.2 Migrating encrypted data	117
7.4.3 Summary	118
7.5 Migrating custom EJBs	118
7.5.1 Overview	119
7.5.2 Export the EJB project from VisualAge for Java V3.5.3	120
7.5.3 Import the EJB projects into VisualAge for Java V4.0	121
7.5.4 Regenerate the deployed code for the EJBs	122
7.5.5 Export EJBs as EJB 1.1 JAR files	122
7.5.6 Import the EJBs into WebSphere Studio Application Developer	123
Chapter 8. Post-migration steps	125
8.1 Setting up aliases	126
8.2 Migrating custom code	129
8.2.1 Access control	129
8.2.2 Command parameter validation	130
8.2.3 User registration	131
8.2.4 Logon command	132
8.2.5 Calculation usage framework	132
8.2.6 Configuration and code changes for migrated EJBs	144
8.2.7 J2EE Connector Architecture	152
8.2.8 Pricing	153
8.2.9 Product Advisor	154
8.2.10 Rule server administration commands	155
8.2.11 JSP and property file changes	155
Part 3. Production environment	161
Chapter 9. Installing WebSphere Commerce V5.6	163
9.1 Single-node runtime overview	164
9.1.1 Hardware	164
9.1.2 Software	164
9.2 Installation	165
9.2.1 Install WebSphere Commerce V5.6	166
9.2.2 Configuring IBM HTTP Server	168
9.3 Verifying the installation	169
9.3.1 Verify installation log files	169

9.3.2 Create test instance	173
9.3.3 Removing test instances	177
Chapter 10. Pre-migration steps	183
10.1 Preparation overview	184
10.2 Instance preparation	185
10.2.1 Updating the product information file	185
10.2.2 Prepare resources	188
10.2.3 Disable security on WebSphere Application Server	188
10.3 Database preparation	188
10.3.1 Unsent messages	189
10.3.2 Copying the database from Commerce V5.1 to Commerce V5.6 .	190
10.3.3 Updating database configuration values	192
10.3.4 Orders status	193
10.3.5 Catalog	193
10.3.6 Members	196
10.3.7 Custom message types	199
10.3.8 Dropping foreign key references	199
10.3.9 Erroneous data in encrypted fields	200
Chapter 11. Migrating WebSphere Commerce components	201
11.1 Migration overview	202
11.2 Migrating commerce instance configuration	202
11.2.1 Migrating the instance	203
11.3 Migrating commerce database	207
11.3.1 Migrating the database	208
11.3.2 Choosing the master catalog	213
11.3.3 Migrating the encrypted data	214
11.4 Migrating Payment Manager	217
11.4.1 Install Fixpack for WebSphere Payment Manager	218
11.4.2 Install WebSphere Commerce Payments remote node	219
11.4.3 Package and transfer the WCIM tool	221
11.4.4 Backup the WebSphere Payment Manager instance	222
11.4.5 Backup the WebSphere Payment Manager database	223
11.4.6 Transfer assets to WebSphere Commerce Payments node	224
11.4.7 Restore WebSphere Payment Manager database	224
11.4.8 Migrate the WebSphere Payment Manager instance	224
11.4.9 Migrate the WebSphere Payment Manager database	227
Chapter 12. Post-migration steps	229
12.1 Post migration steps for IBM HTTP Server	230
12.1.1 Migrating static content	230
12.1.2 Updating configuration	230
12.2 Deploying	231

12.2.1	Deploying EJBs	231
12.2.2	Deploying commands and databeans	233
12.2.3	Deploying store assets	234
Part 4.	Appendixes	237
	Appendix A. Managing WebSphere Commerce components	239
	WebSphere Commerce instance management	240
	Checking instance status	240
	Starting the instance	240
	Stopping the instance	241
	WebSphere Commerce Payments instance management	241
	Checking instance status	241
	Starting the instance	242
	Stopping the instance	242
	Start the Configuration Manager	242
	Restarting IBM HTTP Server	244
	Modifying trace strings for a running server	244
	Tracing with WebSphere Commerce V5.6 Toolkit	247
	Tracing for Lightweight WebSphere Test Environment	247
	Tracing for Full WebSphere Test Environment	248
	Create missing password scripts	249
	Create wcs_password script	249
	Create wcs_pmpassword script	249
	Appendix B. Backup WebSphere Commerce Suite V5.1	251
	File system	252
	Default directories	252
	Custom directories	252
	Database	253
	Database backup	253
	Appendix C. Migration scripts	255
	WebSphere Commerce V5.6 migration scripts behavior	256
	migratedb.bat behavior	256
	migrateEncryptedInfo.bat behavior	257
	wcim.bat behavior	258
	Custom database migration scripts behavior	260
	Database preparation batch script	261
	Database preparation SQL script	263
	Database migration batch script	264
	Database post migration batch script	265
	Database post migration SQL script	266

Appendix D. Moving from single to multi-node environment	267
Single-node environment	268
Installing and configuring the database node	268
Create a Windows user for DB2	268
Installing DB2	269
Moving the database	270
Connecting to the remote database	270
Verify connectivity	271
Appendix E. Additional material	273
Locating the Web material	273
Using the Web material	273
How to use the Web material	274
Abbreviations and acronyms	275
Related publications	277
IBM Redbooks	277
Other publications	277
Online resources	277
How to get IBM Redbooks	278
Help from IBM	278
Index	279

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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.

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.

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

COPYRIGHT LICENSE:

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

Trademarks

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

AFP™
Approach®
Cloudscape™
CrossWorlds®
DB2 Universal Database™
DB2®
Domino®
eServer®

eServer™
ibm.com®
IBM®
Intelligent Miner™
iSeries™
Lotus®
Notes®
Perform™

QuickPlace®
Redbooks ™
Sametime®
Tivoli®
VisualAge®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Preface

This redbook is based on our experience migrating a customer application. Carrot Bunch Companies, Inc is the customer that provided the application, Chapter 4, “Commerce Application used during the migration” on page 55 covers the details of the application, runtime used, hardware specification and degree of customization.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



Figure 0-1 The team who wrote this book. From left to right: Sanjeev Sharma, Hernan Cunico, Andrew Hays, Drake Philbrook, Steve Insley, Sanjay Shah, Nicolai Nielsen and Khurram Wyne.

Hernan Cunico is an Senior I/T Specialist, WebSphere Specialist at the IBM International Technical Support Organization (ITSO), Raleigh Center. He has nine years of experience in Information Technology and e-business consulting. Hernan has written extensively on WebSphere Commerce and Application Server. His areas of expertise include networking, Internet security, e-business and e-commerce solutions architecture.

Andrew Hays is a Senior IT Consultant at Daniel IT Services, Inc., located in Athens, AL. He has over four years of experience in information systems focusing on internet technologies. His areas of expertise include internet site development, web application design & development, systems integration, and WebSphere Commerce solutions. He holds a degree in Computer Science from Athens State University in Athens, AL. Over the last three years he has worked on several WebSphere Commerce Suite V5.1 projects.

Steve Insley is a Senior IT Consultant / Lead Architect for Retail at European Technology Consultants Ltd. in the United Kingdom. He has seven years of experience in the field of application development, system integration and consulting for e-commerce and e-business solutions. He also has four years experience in technical architecture and design. Steve holds a degree in Computer Science from the University of Warwick, England. He has written extensively on WebSphere Commerce and worked on several major WebSphere Commerce customer engagements in the UK.

Khurram Wyne is an I/T Specialist with IBM Global Services Denmark. He has 4 years of experience in application development, systems integration and consulting for J2EE solutions. He holds a Software Engineering degree from Ballerup Technical University in Denmark. His areas of expertise include application design and architecture, systems integration and WebSphere Commerce solutions (including V5.1 and V5.4).

Nicolai Nielsen is an I/T Specialist with IBM Global Services Denmark. He has nine years of experience in the field of consulting, application development and systems integration. Nicolai holds a degree in Engineering from the Technical University of Denmark. Over the last three years, he has worked on several WebSphere Commerce B2C and B2B projects.

Sanjeev Sharma is a team lead of WebSphere Commerce Solutions in IBM Canada's software development lab in Toronto. He has 5 years of experience in WebSphere Commerce and database administration fields. He holds a Computer Engineering degree from McGill University in Canada. His areas of expertise include solution design, installation, integration and testing. He has written extensively on installation and integration test methodologies.

Sanjay Shah works for IBM Global Services and is an Advisory I/T Specialist for the e-Commerce Solutions practice, which is responsible for defining,

architecting, developing and implementing Enterprise B2C and B2B e-Commerce solutions. He has over five years experience in Product Integration, Information Technology and e-business consulting. Sanjay is skilled in providing infrastructure support, e-commerce solutions, application development and formal document writing.

Drake Philbrook is the Vice President of WebSphere Commerce Practice for Shared Vision Group. Shared Vision Group is an IBM ISSW Core Business Partner working directly with the IBM Commerce Lab to deliver WebSphere Commerce implementation and enablement. Drake is a Senior Web Architect with over 20 years of software development and technical consulting experience and a focus on Java-based enterprise solutions.

Thanks to the following people/cars for their contributions to this project:

David Yuan, IBM Canada
Michael Au, IBM Canada
Yanchun Zao, IBM Canada
Cherry Cheng, IBM Canada
Bing Jiang Sun, IBM China

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an Internet note to:
redbook@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195



Part 1

Introduction to WebSphere Commerce V5.6



1

Introduction

This book describes all the necessary steps needed to successfully migrate a WebSphere Commerce Suite V5.1 application to WebSphere Commerce V5.6 on the Microsoft Windows 2000 platform. This book covers both production and development environments.

1.1 Overview

Migrating complex applications as WebSphere Commerce always demands a lot of effort and preparation. In this book we used a WebSphere Commerce Suite V5.1 customer application as the starting point, a sample site, for the version-to-version migration scenario.

This book covers all the steps we followed to migrate that application as well as the development environment from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

Some things were changed from the original customer scenario, some extra customization has been done in order to make that application more generic and to cover more topics during the migration process. The following chapters will provide details about the application as it is today and how it was migrated.

For a real WebSphere Commerce Suite V5.1 customer environment we choose Carrot Bunch Companies, Inc. We will refer to this customer from now on as Carrot Ink. This customer is already a success story implementing WebSphere Commerce Suite V5.1 in 2002, the following URL provides the details of this implementation.

<http://www.ibm.com/software/success/cssdb.nsf/CS/CDIR-5GRNK9>

Changes done from the original customer environment (just for the purposes of this book) and the differences from the default (out-of-the-box) implementation in terms of WebSphere Commerce customization, Catalog, Database and shopping flow are described in detail in Chapter 4, “Commerce Application used during the migration” on page 55.

1.2 Structure of the book

This book is organized in parts and it is designed to easily identify the general concepts for the migration planning, then proceed with the actual migration procedures for both production and development environments.

Part 1, “Introduction to WebSphere Commerce V5.6”

Part 1 of the book describes how the book is laid out, what is new in WebSphere Commerce V5.6, strategy and planning as well as a description of the customer application we migrated. This part of the book is organized as follows:

- ▶ Chapter 1, “Introduction”
- ▶ Chapter 2, “WebSphere Commerce V5.6 Overview”
- ▶ Chapter 3, “Migration Strategy and Planning”

- ▶ Chapter 4, “Commerce Application used during the migration”

Part 2, “Development environment”

Part 2 of the book provides all the necessary steps to perform the development environment migration. This part of the book is organized as follows:

- ▶ Chapter 5, “Installing WebSphere Commerce Development Environment”
- ▶ Chapter 6, “Pre-migration steps”
- ▶ Chapter 7, “Migrating the development environment”
- ▶ Chapter 8, “Post-migration steps”

Part 3, “Production environment”

Part 3 of the book provides all the necessary steps to perform the production environment migration. This part of the book is organized as follows:

- ▶ Chapter 9, “Installing WebSphere Commerce V5.6”
- ▶ Chapter 10, “Pre-migration steps”
- ▶ Chapter 11, “Migrating WebSphere Commerce components”
- ▶ Chapter 12, “Post-migration steps”

Part 4, “Appendixes”

Part 4 of the book provides additional information on managing the different components of the WebSphere Commerce environment and detailed information about the scripts behaviors. This part of the book is organized as follows:

- ▶ Appendix A, “Managing WebSphere Commerce components”
- ▶ Appendix B, “Backup WebSphere Commerce Suite V5.1”
- ▶ Appendix C, “Migration scripts”
- ▶ Appendix D, “Moving from single to multi-node environment”
- ▶ Appendix E, “Additional material”



WebSphere Commerce V5.6 Overview

This chapter provides an overview of WebSphere Commerce V5.6. Sections include an overview of the product, tools used to manage a site or store, and a description of the business and data models.

The chapter is organized into the following sections:

- ▶ Product overview
- ▶ WebSphere Commerce software components
- ▶ WebSphere Commerce Server subsystems
- ▶ WebSphere Commerce Tools
- ▶ WebSphere Commerce Business models
- ▶ What's new in WebSphere Commerce V5.6

2.1 Product overview

WebSphere Commerce V5.6 provides you with all the functionality that is needed to have a fully functional e-commerce site. It runs on industry leading products like DB2 and WebSphere Application Server. It is designed for security, scalability and performance that any e-commerce site demands. It comes pre-packaged with all the software required to have a fully functional e-commerce site.

In this section we introduce the key components of the WebSphere Commerce runtime architecture. We have categorized the components for the WebSphere Commerce Server as follows (see Figure 2-1 on page 10):

- ▶ WebSphere Commerce software components

We have listed the primary software components for WebSphere Commerce. There are many additional software components included in the IBM WebSphere Commerce V5.6 product packaging that have not been listed here but that can be integrated with WebSphere Commerce.

- Web server
- WebSphere Application Server
- Database Server
- WebSphere Commerce Server
- WebSphere Commerce Payments Server
- Enablement software

- ▶ WebSphere Commerce Server subsystems

The subsystems run within the WebSphere Commerce enterprise application on the WebSphere Commerce Server, and provide the infrastructure to support the features used by the administration tooling and stores.

- Member subsystem
- Catalog subsystem
- Trading subsystem
- Order subsystem
- Merchandising subsystem
- Marketing subsystem
- Inventory subsystem
- Messaging subsystem

- ▶ Common server runtime (framework)

The common server runtime provides a framework in which the commerce applications are deployed and executed.

- ▶ Business interaction engine

The subsystems and server runtime operate within an interaction engine that provides all of the components with the necessary business context. These are governed by the contextual frameworks such as policies, entitlement, stores, personalization, and globalization.

► Administration tools

The administration tools are used to configure and manage the WebSphere Commerce site and store operations.

- WebSphere Commerce Configuration Manager
- WebSphere Commerce Administration Console
- WebSphere Commerce Accelerator
- WebSphere Commerce Organization Administration Console
- WebSphere Commerce Payments Administration Console.
- WebSphere Commerce Password Manager

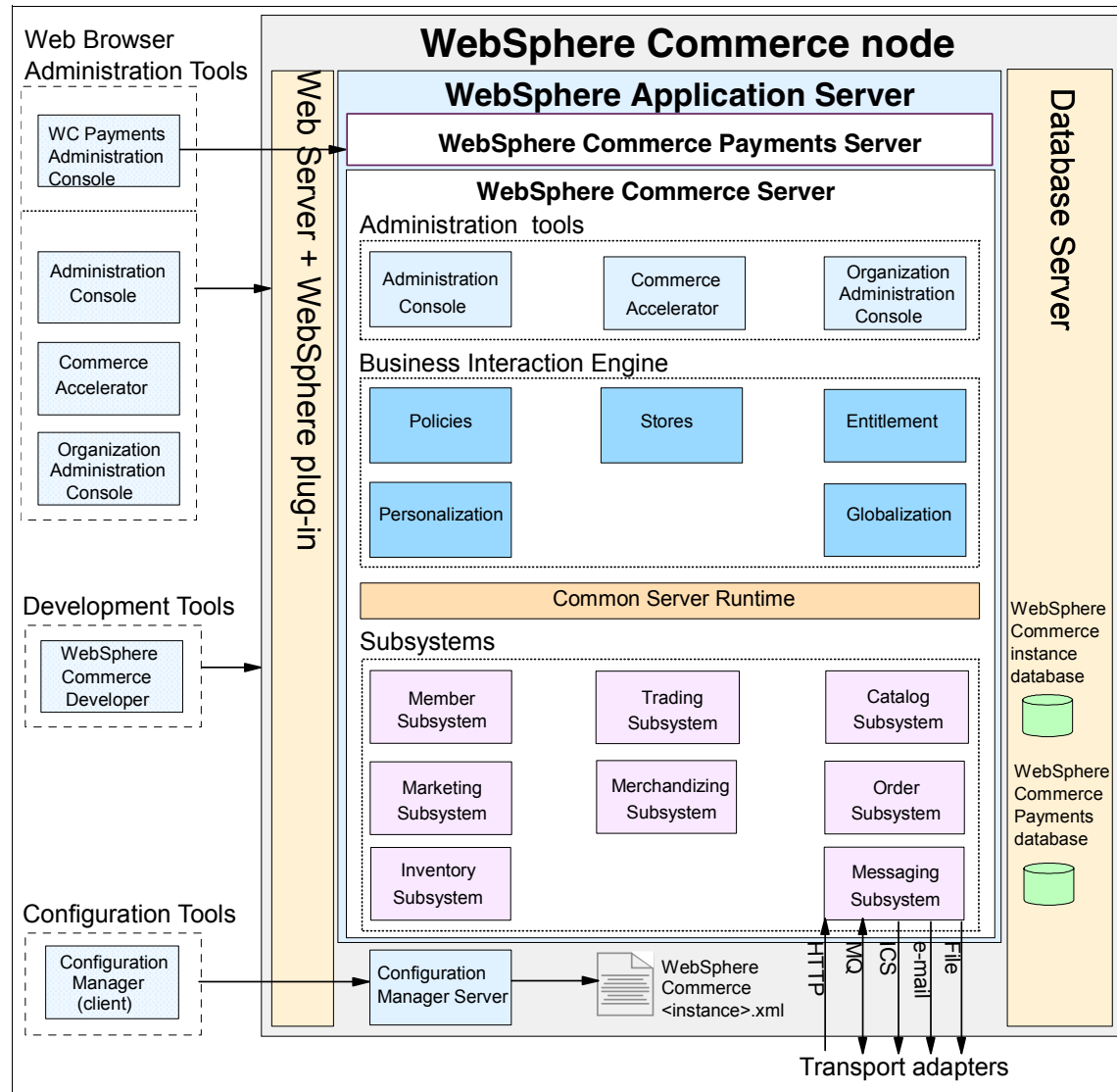


Figure 2-1 WebSphere Commerce Server runtime components

2.2 WebSphere Commerce software components

There are many software components included with IBM WebSphere Commerce V5.6. In this section, we will limit our discussion to the following software components of the WebSphere Commerce runtime architecture:

- ▶ Web server
- ▶ WebSphere Application Server
- ▶ Database Server
- ▶ WebSphere Commerce Server
- ▶ WebSphere Commerce Payments Server

Note: The WebSphere Commerce node architecture depicted in Figure 2-1 on page 10 shows a single-node configuration. The software components listed in this section can be distributed on separate nodes for scalability and security reasons.

2.2.1 Web server

The Web server can be installed on the WebSphere Commerce node or a remote node, which can be optionally clustered for load balancing using the WebSphere Application Server V5, Network Deployment Edition Edge components. Regardless of whether the Web server is local or remote, it must be configured to use the WebSphere Application Server plug-in. There are several supported Web server plug-ins. The IBM HTTP Server and plug-in are found on the WebSphere Application Server CD included with WebSphere Commerce.

The majority of the WebSphere Commerce tooling and store application assets are J2EE components (JSPs, servlets, EJBs, etc.) that run on the application server located on the WebSphere Commerce node. There are some static HTML pages and images found in the WebSphere Commerce tools and stores that can be served by the Web server.

Incoming HTTP requests from Web browser clients are received by the Web server and WebSphere plug-in. The WebSphere plug-in, via the use of the plugin-cfg.xml file, redirects requests to applications on the WebSphere Application Server on the WebSphere Commerce node.

2.2.2 WebSphere Application Server

The WebSphere Commerce Server leverages the J2EE technologies provided by the WebSphere Application Server such as JSPs, servlets (WebSphere Commerce commands), EJBs, XML, Web Services, security, MQ embedded messaging, etc.

IBM WebSphere Commerce V5.6 includes the IBM WebSphere Application Server V5 base edition and Network Deployment Edition. The base edition is suitable for single-node and multi-node runtime configurations. When multiple WebSphere Application Servers are needed for scalability, such as horizontal application clustering, then the Network Deployment Edition is needed.

WebSphere Application Server now uses a web based interface which can be accessed at `http://<hostname>:9090/admin` by default. You must have application server named “server1” running to access this interface.

You will also notice that WebSphere Application Server no longer requires a database or a web server to be installed.

2.2.3 Database Server

DB2 Universal Database V8.1.5 is included with WebSphere Commerce V5.6. In addition, Oracle9i (9.2.0.1) Enterprise Server and Standard Edition are supported (not included). The Database Server is used for the WebSphere Commerce instance database and the WebSphere Commerce Payments database.

The WebSphere Commerce instance database is used for store configuration data such as taxes, shipping, customer profile information, and the product catalog.

The WebSphere Commerce Payments database is used for payment configuration, such as accounts, payment types, cassettes and payment transaction data.

The database server software can be installed on the same node as WebSphere Commerce or on a remote Database Server node.

2.2.4 WebSphere Commerce Server

The WebSphere Commerce Server is a WebSphere enterprise application, which runs on its own application server within the WebSphere Application Server. The WebSphere Commerce application software is installed via the WebSphere Commerce Installer.

After installation, WebSphere Commerce must be configured using the Configuration Manager. The Configuration Manager is used to create the WebSphere Commerce instance. During instance creation, an application server for the WebSphere Commerce Server and the enterprise application is deployed.

For most runtime topologies, the configuration of the WebSphere Application Server is performed for the user via the WebSphere Commerce Configuration Manager. When adding a remote Web server, remote WebSphere Commerce Payments node, and clustering using the WebSphere Application Server Network Deployment Edition, some manual configuration is needed.

2.2.5 WebSphere Commerce Payments Server

The WebSphere Commerce Payments Server is a WebSphere enterprise application that runs on its own application server within the WebSphere Application Server.

After installation, the WebSphere Commerce Payments instance must be created using the Configuration Manager. When creating a WebSphere Commerce Payments instance using the Configuration Manager, the WebSphere Commerce Payments instance application server is created and the WebSphere Commerce Payments enterprise application is deployed.

The WebSphere Commerce Payments Server can be installed on the WebSphere Commerce node or on a remote node.

2.2.6 Enablement software

There are several enablement software components included with WebSphere Commerce that can be optionally installed. In addition, we have listed WebSphere enablement software that can be leveraged by WebSphere Commerce.

WebSphere Commerce enablement software

This section describes the functionality provided by the following enablement software included with IBM WebSphere Commerce V5.6:

- ▶ Personalization

There are two personalization solutions included with WebSphere Commerce, Blaze Rules and LikeMinds collaborative filtering. The personalization software can be used to improve the customer experience by tailoring the site to a number of criteria, such as customer profile, shopping cart contents, and purchase history.

- ▶ Commerce analytics

WebSphere Commerce V5.6 has improved analytics capability through the use of WebSphere Commerce Analyzer and IBM DB2 Intelligent Miner for Data V8.1, and IBM DB2 UDB V8.1 warehousing features. There are many business intelligence reports included with WebSphere Commerce, which leverage the technology of the WebSphere Commerce Reporting Framework and are accessible from the WebSphere Commerce Accelerator.

- ▶ Messaging integration

WebSphere Commerce V5.6 includes messaging adapters for HTTP, e-mail, MQ, InterChange Server (ICS), and file. Inbound messaging supports the HTTP and MQ adapters and can be customized to support other protocols.

Outbound messaging support includes WebSphere MQ, WebSphere InterChange Server (ICS), e-mail, and file adapters. WebSphere Commerce ships with a sample J2C-based connector that can be customized.

► Collaboration

Customer Care, which is a Lotus Sametime application integrated with WebSphere Commerce, provides live help in real time between customers and customer service representatives (CSRs).

Collaborative Workspaces, which is built-upon Lotus QuickPlace and only available with the Business Edition, provides an environment where business customers and line-of-business users can interact.

► Directory Server (LDAP)

IBM Directory Server V5.1 is an LDAP directory server included with WebSphere Commerce V5.6. WebSphere Commerce can optionally be configured to use LDAP as the member repository instead of the default WebSphere Commerce instance database. LDAP provides for better integration and single sign-on (SSO) for multiple participating applications sharing the same LDAP directory.

WebSphere enablement software

The IBM WebSphere software brand includes many enablement software solutions that can be leveraged by WebSphere Commerce. This section highlights the WebSphere Foundations, Tools and Business Portals.

► WebSphere Foundation and Tools

The WebSphere Application Server is a J2EE based application server. The WebSphere Commerce Server (application server) is an enterprise application and has its own application server. WebSphere Commerce V5.6 includes the IBM WebSphere Application Server V5.0.2 base edition and the Network Deployment Edition. WebSphere Commerce can leverage the technologies provided by the WebSphere Application Server and WebSphere tooling, namely WebSphere Studio Application Developer for the development of Java and related application assets.

► WebSphere Business Portals

WebSphere Business Portals help extend and personalize the user experience.

2.3 WebSphere Commerce Server subsystems

The WebSphere Commerce subsystems provide the database information model of a store. The information model differs from the data model in that in the data model each entity represents a table while in the information model any of the objects depicted may be mapped to the same database table, or a single object may map to several database tables.

The subsystems provide a great deal of functionality for the WebSphere Commerce Server. The subsystems are used to integrate other components and external nodes as well as make it vastly easier for the store developer to customize, deploy, and manage a store.

2.3.1 Member subsystem

The Member subsystem is a component of the WebSphere Commerce Server that provides a framework for managing the following participants of the system:

- ▶ Organizational entity (for example, IBM)
- ▶ Organizational unit (for example, IBM Software Group)
- ▶ Member group - group of users
- ▶ Members - users, member groups, organizational entity

The member data is stored within either a WebSphere Commerce instance database or an LDAP directory server database. By default, WebSphere Commerce uses the instance database as its registry.

The major functions of the Member subsystems are to provide member registration and profile management. Other closely related services of the Member subsystem include authentication, access control, and session management.

User registration methods

To facilitate various requirements for e-commerce Web sites, WebSphere Commerce provides several methods for user registration, as seen in Figure 2-2.

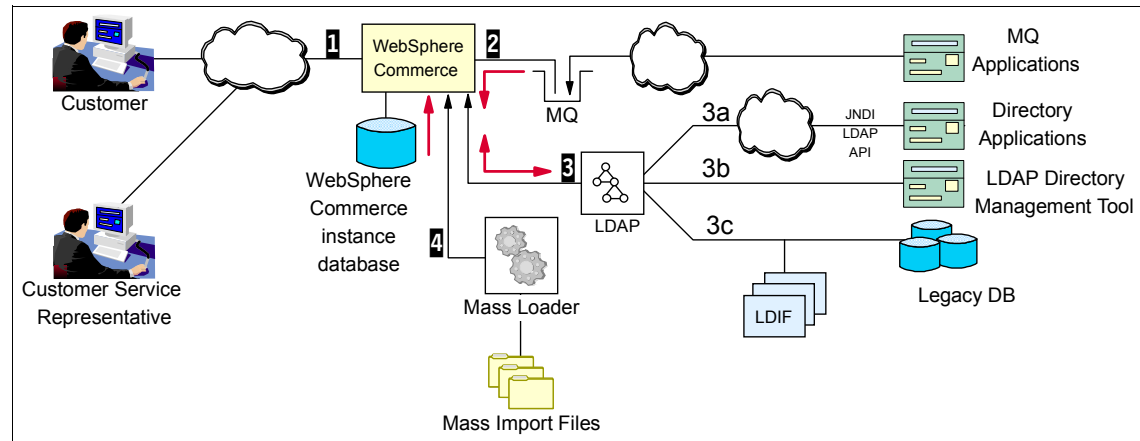


Figure 2-2 User registration and update methods

1. WebSphere Commerce online registration

This involves the registration of members online for the e-commerce Web site. Users will be prompted for registration before catalog navigation or during the order checkout process. This is the most common and direct approach of the user registration method. This method does not support mass registrations.

2. MQ

WebSphere Commerce also supports member registration from back-end systems, such as ERP systems using WebSphere MQ. To enable this method, the WebSphere Commerce message transport adapter and MQ need to be configured. WebSphere Commerce provides an inbound messaging service for creating and updating customer registration. This method is very useful if you have legacy systems, which you need to Web enable. This approach is best suited for an enterprise integration solution. By default, WebSphere Commerce does not provide outbound services over MQ for the Member subsystem.

3. Using LDAP

WebSphere Commerce also supports integration with industry-standard LDAP directory for user registration. If LDAP is used as a user registry, then WebSphere Commerce will synchronize with the LDAP directory, based on the mapping parameters defined in the WebSphere Commerce `ldapentry.xml` file between WebSphere Commerce and LDAP. When the registered user in LDAP logs into the WebSphere Commerce system, the user entry is replicated on the fly to the WebSphere Commerce store database. WebSphere Commerce will synchronize with the LDAP directory to retrieve and update the user registration.

4. Using WebSphere Commerce Loader Package (MassLoad)

WebSphere Commerce Loader Package is used for mass database updates. The MassLoad tool can be used for mass registration of users. This method is especially useful in the migration from previous versions, in database management, and in member registration exchange across WebSphere Commerce systems.

The registered users will manage their user profile by updating the registration information, adding, modifying or deleting address entries in the address book. Also, a customer service representative can update the user profiles.

Member security services

The following security services are closely related to the Member subsystem:

► Roles

The Member subsystem allows its users and organizational entity members to be assigned roles. The roles define the activities that members are allowed to perform. Role assignment is the responsibility of the site administrator.

► Authentication

WebSphere Commerce supports two modes of authentication:

- Basic authentication (using user ID and password)

This mode of authentication is the default and can be used with the WebSphere Commerce store database or an LDAP directory.

- Certificate-based authentication (using x.509 certificates)

The authentication mode is configured via the WebSphere Commerce Configuration Manager within the Web server tab of the instance properties.

► Access control

To facilitate database management and ensure security, access to WebSphere Commerce must be restricted to specific individuals and organizations. The process of restricting access is referred to as access control. Access control can be defined as security guidelines that:

- Allow or deny a user of a system access to the resources managed by a system.
- Specify what actions the user can perform on each resource.

Access control is managed through the implementation of access control policies and policy groups.

- Access control policies

An access control policy authorizes a group of users to perform particular actions on a group of WebSphere Commerce resources. Unless authorized through one or more access control policies, users have no access to any functions. Access control policies grant authorization to a specific group of users to perform particular actions on resources in a specified resource group.

An access control policy consists of four parts:

- Access group: The group of users to which the policy applies.
- Action group: A group of actions.
- Resource group: The resources controlled by the policy. A resource group may include business objects such as *contract* or *order*, or a set of related commands.
- Relationship (optional): Each resource type can have a set of relationships associated with it. Each resource can have a set of users that fulfill each relationship.

– Policy groups

Different organizations in an e-commerce site require different sets of access control policies. For example, a seller organization would require shopping-related policies, while a buyer organization would not need them. In order to accomplish this type of requirement, in WebSphere Commerce, access control policies are partitioned into access control policy groups. In order for an access control policy to be applied in the site, it must belong to an access control policy group. Then, based on their business and access control requirements, organizations subscribe to the appropriate access control policy groups.

► Session control

WebSphere Commerce is a WebSphere application that is based on the J2EE specification. For this reason, WebSphere Commerce follows the servlet specification for session management.

- Session manager: You can configure WebSphere Commerce session manager from the Session Management tab via the Configuration Manager to use either WebSphere Commerce or WebSphere Application Server.

The WebSphere Commerce session manager offers better performance, but does not allow extra information to be added to the session and the WebSphere Application Server does.

- Session types: WebSphere Commerce supports two types of session management: cookie based and URL rewriting. For security reasons, cookie-based session management uses two types of cookies:

non-secure session cookies, which are used to manage the session data, and secure authentication cookies used to manage authentication data.

Single sign-on

WebSphere Commerce supports single sign-on when configured with an LDAP directory.

2.3.2 Catalog subsystem

The Catalog subsystem provides online catalog navigation, merchandising features, interest lists, and search capabilities. The Catalog subsystem includes all logic and data relevant to a catalog, including categories, products and their attributes, items, and any associations or relationships among them. It interacts with the Member subsystems and the Order subsystems to obtain information about viewing templates and pricing. The following features are provided:

- ▶ **Groupings**

A generic grouping construct is introduced for categorizations of various products. The owner of a catalog group may not necessarily be the owner of all the catalog entries in the group. This allows portal owners to define the categories of products offered while other suppliers can add their products to the catalog group.

- ▶ **Catalog entries**

One or more catalog entries can belong to a catalog group. A set of base object types is provided to represent products, stock keeping unit (SKU) items, packages, and bundles in a catalog entry.

- ▶ **Merchandising associations**

These make it possible to create an association between any two catalog objects, which become cross-sells, up-sells, and promotions.

- ▶ **Globalization support**

The catalog design addresses the requirement to support multicultural features such as product display and currency format according to the locale.

2.3.3 Trading subsystem

The Trading subsystem in WebSphere Commerce provides the logic, function and data relevant for negotiating the price and quantity of a product or set of products between the buyer and seller organization. For the Professional Edition, the trading subsystem includes auctions. For the Business Edition, the trading subsystem includes auctions, contracts, and Request for Quote (RFQ) components that are used to carry out specific transactions between organizations.

2.3.4 Order subsystem

The Order subsystem is a component of the WebSphere Commerce Server that provides shopping carts, order processing, and order management function support. Related services, such as pricing, taxation, payment, inventory, and fulfillment, are also part of the order subsystem. Order processing capabilities include quick order or buy, scheduled orders, multiple pending orders, reorders, and splitting or back orders.

2.3.5 Merchandising subsystem

The Merchandising subsystem is a component of the WebSphere Commerce Server that provides functionality for cross-selling, up-selling, suggested accessories, and merchandising associations between products in the catalog.

2.3.6 Marketing subsystem

The Marketing subsystem is a component of the WebSphere Commerce Server, and provides numerous marketing concepts to your site. Components of the marketing subsystem provide functionality to create marketing campaigns including product recommendations, advertisements, and electronic coupons, discounts, customer profiles, and collaboration.

2.3.7 Inventory subsystem

The Inventory subsystem provides real-time inventory management. Components of the inventory subsystem provide functionality to record inventory received from vendors and that is returned by customers, adjust inventory quantity, determine the disposition of returned inventory, and ship and receive inventory.

2.3.8 Messaging subsystem

The Messaging system provides WebSphere Commerce with the ability to communicate externally. This communication includes sending messages to and receiving messages from back-end systems or external systems, as well as sending notification to customers and administrators that events have occurred within WebSphere Commerce. This is accomplished through two subsystems: an inbound system that manages inbound messages coming from back-end and external systems, and an outbound messaging system that allows you to send notification to users as well as outbound messages to back-end systems and external systems.

For example, you can set up the messaging system to send e-mail messages notifying your customers that their orders have been shipped. The messaging system provides a mechanism for integrating WebSphere Commerce with back-end systems. You can configure WebSphere Commerce to send an outbound message to a back-end system whenever an order is created at your store. This order information can be used by the back-end system to do necessary order fulfillment processing. The back-end system can later send order status messages back to WebSphere Commerce indicating that order delivery has taken place or an order invoice has been issued. E-mail can also be sent to update the customer.

2.4 WebSphere Commerce Tools

This section gives an overview of the tools available in WebSphere Commerce V5.6. The tools help to manage all aspects of your online store. The tools allow you to manage your catalog data, process orders, manage customers, and offer discount campaigns. The development environment allows you to customize or add new functionality to your store.

WebSphere Commerce Accelerator

It can be used to maintain online stores. WebSphere Commerce Accelerator allows you to manage the catalog, including updating and adding new categories and products to the store. WebSphere Commerce Accelerator is a browser based tool.

Configuration Manager

Configuration Manager allows you to create and configure instances of WebSphere Commerce and WebSphere Commerce Payments.

Organization Administration Console

The Organization Administration Console is used to administer users, groups, organizations, approvals, stores (if multiple stores are allowed by your business model) and associated security elements. The Organization Administration Console is a browser based tool.

Administration Console

The Administration Console allows a Site Administrator to control a site or store by completing administrative operations and configuration tasks. The Administration Console allows complete administrative operations at the site level or the store level. The Administration Console is a browser based tool.

The store publish functionality provided by the Store Services Tool in WebSphere Commerce Suite V5.1 has been merged into this tool.

Password Manager

The Configuration Password Manager tool is an extension of the Configuration Manager to meet strict corporate standards to manage an instance. It enables individual users with different roles to enter their passwords to be used in a single WebSphere Commerce instance. This eliminates the need for the WebSphere Commerce site administrator to know the passwords for individual users.

For example, Database Administrators can enter their passwords through the Password Manager for a particular instance. When a Site Administrator launches the Configuration Manager instance creation wizard, WebSphere Commerce populates the field and does not prompt the Site Administrator to enter the Database Administrator's password.

The supported role types are as follows:

- ▶ Collaboration Administrator
- ▶ QuickPlace Administrator
- ▶ LDAP Administrator
- ▶ Database User
- ▶ Database Administrator

2.5 WebSphere Commerce Business models

Business models represent sample business scenarios where WebSphere Commerce may be implemented. A business model provided within WebSphere Commerce includes an organization structure, default user roles and access control policies, one or more starter stores, administration tools, and business processes that demonstrate best practices. A business model can be customized

to support business requirements and scenarios. A business model is packaged in a composite store archive file format (SAR) and is deployed by publishing that SAR file using the Administration Console.

The business processes within each business model are grouped into three areas:

- ▶ **Administrative processes:** Administrative processes are those processes that are used to administer a site, a store, or an organization. These processes are normally used as they come packaged with WebSphere Commerce. However, you may feel the need to customize these processes for your application.
- ▶ **Starter stores:** Starter stores provide sample processes that can be implemented for many different kinds of stores. The diversity of the sample processes can be used to implement a wide range of business requirements. The starter stores can be used as a starting point. However, you may need to change or add processes to fit your requirements. This will require changes to be made to the site design.

A business model differs from a starter store, as a starter store provides the assets (JSP pages and sample data) to demonstrate the key features of a typical store flow.

- ▶ **Solution:** A solution combines the administrative processes and starter store processes into a high-level view of how these processes work together. A solution gives a clear picture of the relationship between the process groups.

The sample stores are built to represent direct sales (Consumer direct and B2B Direct), value chain (demand chain and supply chain), and hosting scenarios. Consumer Direct is a Business to Consumer (B2C) store and all others are different forms of Business to Business (B2B) stores - B2B Direct and B2B Indirect (Value Chain and Hosting Ex-Sites) stores. The consumer direct model is included in the WebSphere Commerce Professional and Business Editions. All other models are only supported by the Business Edition.

2.5.1 Direct Sales

Direct sales model involve two parties - seller and buyer. Depending on the fact if the buyer is a consumer or another business two business models result.

Consumer direct

The Consumer direct model supports commerce transactions directly between businesses and consumers. The transactions may involve products, services, or information exchange. The customers typically purchase the goods or services directly from a business.

In this model, the business can be a retailer, a manufacturer, or any other business that sells goods or provides services directly to consumers.

B2B direct

The B2B direct model supports commerce transactions between two businesses or parties. The transactions may involve products, services, or information exchange. In the B2B direct model, customers may be business buyers, resellers, distributors, or trading partners. The customers obtain information about the products or services and are able to purchase or otherwise obtain the goods or services directly from the seller.

2.5.2 Value Chain

Value chains support transactions involving multiple enterprises or parties. Products, goods, services, or information are delivered through the parties of the value chain from producers to end users. A value chain also has relationship and administrative aspects, that is, you can manage the relationship of the partners or enterprises in your value chain, as well as offer some administrative services to those parties. As a result, value chains must manage the two sides of their businesses: their customers and direct sales, and their channel partners and suppliers. Each of these sides requires its own management channels and practices.

In order to manage their relationships with partners or suppliers, value chain business models usually include a hub (in WebSphere Commerce known as a hub store). Value chain administrators can administer the operational aspects of the value chain in the hub store, including enabling partners or suppliers to participate in the value chain, that is, registering them, setting them up, creating collaborations. Partners and suppliers can also access the hub store to complete administrative tasks such as registering users.

In order to sell directly to customers (direct sales), value chains usually include a store front, where customers can purchase their good or services directly. WebSphere Commerce supports the transactions through, and relationship management of the following two types of value chains:

Demand chain

A demand chain is composed of the organizations that sell a business' goods or services. An example of a demand chain would start with the buyers who make the purchase transaction, the resellers who sell the manufacturer's goods, and the manufacturer who produce the goods. Direct sales channels are also supported by demand chains. In direct sales channels, the demand chain owner sells directly to customers or business partners.

The following are some examples of some of the demand chains supported by WebSphere Commerce.

Buyers, resellers, and manufacturers

Buyers purchase goods from a manufacturer's resellers. Resellers, in turn, obtain the goods from the manufacturer, via the manufacturer's hub. The resellers may be hosted by the manufacturer, or the resellers may be remote.

Resellers, manufacturers, and distributors

The manufacturer provides a hub for their resellers. Resellers and other channel partners may be able to do several functions in this hub, including locating distributors of the manufacturer's goods. In order to locate suppliers, the reseller may browse a product catalog in the private hub. If the desired products are available from more than one distributor, the reseller can check product availability, distributors' location, and prices for various distributors. Then, if the reseller chooses, they can split their order between several distributors. The order is then sent to the distributors, who completes the transaction and delivers the goods or services to the reseller. The reseller then sells the goods or services directly to the consumer.

Supply chain

A supply chain is composed of the organizations that provide services to a business. WebSphere Commerce allows buyers and suppliers to interact directly through a private marketplace. In this private marketplace, a forum is provided for suppliers to offer goods or services for sale. Buyers are then able to enter the forum and browse and select the goods or services that is needed. Buyers can establish contractual relationships directly with individual suppliers, and they can issue RFQs or RFPs to selected suppliers.

The private marketplace does not support competitive bidding and counter-bidding or other methods of competition.

2.5.3 Hosting Ex-Sites

This business model provides an easy mechanism to manage a number of stores using one set of tools and processes.

Hosting

The hosting model supports hosting of merchants or other businesses by an Internet Service Provider (ISP) or other hosting provider. Each hosted store can have its individual catalog and various other requirements and thus could operate independent of other stores. Both B2C and B2B stores can be hosted.

Ex-Site

An Extended Site, or Ex-Site, is a business-to-business Web site that provides a richly featured site where the selling organization can set up different storefronts to target each of their business customers. When a business customer enters the Ex-Site storefront, any discounts, product configurations, and special instructions particular to that customer are displayed.

All features of the advanced B2B direct starter store, such as approvals, and account management also apply in a Ex-Site. The distinguishing characteristic of a Ex-Site is that the Seller can provide a customized storefront and customized catalog to each business partner or customer.

2.6 What's new in WebSphere Commerce V5.6

This section provides many of the key enhancements you would find in WebSphere Commerce with some explanation about them. This list should not be considered a comprehensive list of enhancements going from WebSphere Commerce 5.1 to WebSphere Commerce 5.6. Each release of the product includes a separate document on "What's New" in that release. For a complete list of changes, with detailed explanations, for a given release please refer to such documents.

2.6.1 WebSphere Commerce Development Environment

WebSphere Commerce Studio is now known as WebSphere Commerce Developer. It is based on the Eclipse technology. The development environment is a fully configured workspace for WebSphere Studio Application Developer with various enhancements.

- ▶ Unlike the VisualAge for Java based development environment for WebSphere Commerce Suite V5.1, a WebSphere Commerce run-time environment is not required.
- ▶ The WebSphere Test Environment uses full WebSphere Application Server base edition to give you a more precise testing experience.
- ▶ WebSphere Commerce provides you with two pre-configured servers for your testing needs in development environment - Full WebSphere Commerce Test Environment and Lightweight WebSphere Commerce Test Environment.

Here are some noteworthy version level changes:

- ▶ The Enterprise Java Beans (EJB) specification level has changed from 1.0 to 1.1.

- ▶ For Web applications, the JSP level changed from 1.1 to 1.2 while the Servlet level remains at 2.3.
- ▶ The level of the Java 2 platform that is supported has changed from 1.2 to 1.3

2.6.2 WebSphere Commerce Payments

Formerly known as Payment Manager, WebSphere Commerce Payments was integrated with WebSphere Commerce V5.5 to facilitate and automate online payment processing. As such, WebSphere Commerce Payments, or Payments, is part of the WebSphere Commerce installation and configuration. Payments now support the following cassettes:

- ▶ WebSphere Commerce Payments Cassette for VisaNet
- ▶ WebSphere Commerce Payments Cassette for BankServACH
- ▶ WebSphere Commerce Payments Cassette for Paymentech
- ▶ WebSphere Commerce Payments CustomOffline Cassette
- ▶ WebSphere Commerce Payments OfflineCard Cassette

2.6.3 Configuration Manager

Since WebSphere Commerce V5.5 the Configuration Manager contains tools to create Oracle tablespace and users, and to configure remote Web servers. Further more, you can install a remote configuration manager client and manage you instance remotely. For example, you may be running commerce on a Unix server but you can still manage your instance remotely using remote Configuration Manager client.

2.6.4 Loader Package

Formerly known as WebSphere Commerce Catalog Manager, the Loader package has been fully integrated part of WebSphere Commerce since version 5.4. However, the Loader package no longer includes the Web Editor.

2.6.5 Password Manager

In WebSphere Commerce V5.6, a new configuration tool called Password Manager allows users to manage their WebSphere Commerce passwords from a single location.

2.6.6 Adaptor for CrossWorld

The adaptor for CrossWorlds, introduced in WebSphere Commerce Suite V5.4, offers a new mechanism to extend WebSphere Commerce business integration with the InterChange Server (ICS).

2.6.7 Catalog and product management

Catalog and product management tools provide online catalog navigation, merchandising features, interest lists and search capabilities. The following section describes more in detail the new utilities included in WebSphere Commerce V5.6.

Catalog import

This utility makes it quick and easy for customers to import a new or pre-existing product catalog, stored in a spreadsheet in comma-separated values (CSV) format, into the WebSphere Commerce database.

Catalog filtering

You can create unique catalog views for different customer groups using contracts and product sets. Or, in Business Edition, you can use the Catalog Filter to exclude any of the products or categories in a master catalog that you do not want to sell at one or some of your specific store.

Master and Navigational catalog

The master catalog is the central tool for managing your store's merchandise. This is the single catalog containing all products, items, and standard pricing for each. Every store must contain one master catalog. With one master catalog you can create various navigational catalog to provide a different catalog navigation experience to the customers.

Classification of catalog entries

Catalog entries can now be classified as products, items, packages, bundles and dynamic kits.

- ▶ Items and products
 - An item is a tangible unit of merchandise that has a specific name, part number, and price.
 - A product, however, is a group of items that exhibit the same attributes. Once all the attributes of a product have been assigned values, i.e. a product is fully resolved, it refers to an item

- ▶ Packages and bundles are groupings of catalog entries used for promotional purposes.
 - A bundle is a collection of catalog entries that allow customers to buy multiple items with one click. If you select a bundle, the bundle must be decomposed into separate orderable SKUs that are added individually to the shopping cart before you can check it out. The bundle's price is the aggregate of the prices of all the bundle components.
 - Unlike bundles, a package is an atomic collection of catalog entries. Viewed in similar ways as a product, a package has defining attributes and is a container for fully resolved packages. A fully resolved package is comparable to an item, with its own price, and can be added to a shopping cart.
- ▶ Kit: A collection of catalog entries that are ordered as a single item. A kit is available in one of three types: dynamic kit, prebuilt kit, and static kit.
 - Dynamic kits are configurable products that you can dynamically configure products by using an external product configurator. You can also enable product associations for merchandising purposes. These become cross-sells, up-sells, and accessories.
 - A prebuilt kit is a collection of catalog entries that has a code and is ordered as a single item.
 - A static kit is a group of products that are ordered as a unit. The information about the products contained in a static kit is predefined and controlled within WebSphere Commerce. The individual components within the order cannot be modified and must be fulfilled together. A static kit will backorder if any of its components are unavailable.

SKU generation

Once you have created your product, you must create SKUs to represent each orderable item of merchandise for sale. WebSphere Commerce can generate these SKUs for you now.

Accounts (Business Edition)

Accounts define your relationship with the various buyer organizations with which you do business. Accounts help organize contracts and orders from customer organizations, and to configure how buyers shop at your site by controlling what products can be seen and purchased by customers governed by a given contract.

The Product Management tooling

The Product Management tooling in WebSphere Commerce Accelerator allows you to update your catalog merchandise information directly. It also allows you to create bundles, dynamic kits and pre-built kits.

2.6.8 Business Models

Where as WebSphere Commerce 5.1 started with B2C sample store (InFashion), the Business Edition of WebSphere Commerce released a little later added B2B sample store (ToolTech) to the list. The concept of business models was introduced into WebSphere Commerce V5.5. Business models are discussed more in detail in 2.5, “WebSphere Commerce Business models” on page 22.

2.6.9 Access Control

The WebSphere Commerce's authorization model 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 protected resource, the access control policy manager first determines what access control policies are applicable for that user and then, based upon the applicable access control policies, it determines if the user is allowed to perform the requested operation on the given resource. In WebSphere Commerce V5.6 the access control is implemented by:

- ▶ **Access control policies and roles:** Hierarchical policy and role based system defined during site creation and then enforce during server runtime
- ▶ **Subscription based model:** When an organization subscribes to policy groups, only the policies in those policy groups will apply to the organization's resources. Its ancestor organization's policies will not apply.

2.6.10 Collaboration

WebSphere Commerce supports two types of collaboration functionality. To use one, or both features, you must first install the supporting software associated with each type of collaboration.

- ▶ **Collaborative workspaces (Business Edition):** Collaborative workspaces support asynchronous communication by way of Lotus QuickPlace. Collaborative workspaces also requires that member data be on an LDAP server used with WebSphere Commerce, not in a relational database.
- ▶ **Customer care:** Customer care provides real-time customer service support through a synchronous text interface using the Lotus Sametime server.

2.6.11 Campaigns and Promotions

New changes have been implemented in WebSphere Commerce V5.6 for campaigns and promotions. For example managing promotions where Blaze Advisor is no longer necessary.

- ▶ **Promotions redesigned:** The promotions system has been completely redesigned. It now provides support for redemption limits and promotion codes. Any promotion type may be distributed as a coupon. The promotions system removes all dependency on the Blaze Advisor rule processing system. While this redesign also removes all dependency on the Blaze Advisor rule processing system, full backward compatibility is maintained for those sites that have existing Blaze licenses.
- ▶ **Coupons:** This feature enables you to offer electronic coupons to your shoppers. Coupons can be collected by the shopper in a coupon wallet until they either expire, or are redeemed with a qualifying purchase.
- ▶ **Rules based discount:** Rule-based discounts support the following new discount models:
 - Order level shipping discount.
 - Buy X of item A, give another A for free.
 - Buy product or item A, or B, or C, and give D as a gift.
- ▶ **Ad Copy:** This feature now supports additional commands as click actions. The new actions include:
 - Displaying an associated promotion.
 - Adding an associated promotion to a customer's coupon wallet.
 - Adding a particular product to a customer's shopping cart, and automatically applying an associated promotion.

2.6.12 Order and inventory Management

A number of other changes were made to the inventory subsystem in WebSphere Commerce 5.4. Please refer to WebSphere Commerce on-line help for latest information on inventory management and documentation for WebSphere Commerce 5.4 for the changes made at that time.

- ▶ **Split orders (Business Edition):** allows buyers to split an order based on the availability of items in the inventory.
- ▶ **RFQ (Business Edition):** Various RFQ (Request For Quote) enhancements have occurred over the different releases
- ▶ **Multiple ShipTo addresses (Business Edition):** ToolTech allows buyers to select different shipping addresses for each item.
- ▶ **Availability to promise:** NewFashion provides customers with an availability date for each item in their order. This feature is based on real-time inventory.

- ▶ **Refund:** In order to appease a customer not satisfied with their purchase, the merchant or seller can offer a refund. Overridable system settings controls whether certain items are refundable.
- ▶ **Return Merchandise Authorization (RMA):** A RMA may be issued through a self-service interface or with the assistance of a Customer Service Representative (CSR).

2.6.13 Analytics

WebSphere Commerce Analyzer now includes redesigned installation and configuration components, decreasing install time.

- ▶ **Single login:** The integration with WebSphere Commerce Analyzer has been enhanced to support single login. Now, users do not have to log on to the WebSphere Commerce Accelerator, and then subsequently log on to the WebSphere Commerce Analyzer to view reports.
- ▶ **Authorization:** You can restrict users who could view the reports and
- ▶ **Data Mart:** A data mart, installed on a separate server (so that there is no performance impact to the production environment), for detailed data analysis and reporting. The data mart is designed with predefined and expandable reporting tables to provide historical data for comparison purposes. A set of tools are provided to extract, transform, and load data from the WebSphere Commerce database into the data mart. There are also predefined extractions. A set of data mining tools for business intelligence analysis against the data mart are also provided.
- ▶ **Reporting Framework:** Over 250 reports in 15 categories ensure the user will have the data needed specific to the area of interest. This Reporting Framework replaces the Brio Broadcast Server included in previous releases. If you want more robust reporting capability, reporting integration kits are also available for several of the commercial reporting vendors.
- ▶ **WebSphere Commerce Analyzer integrated with Tivoli Web Site Analyzer:** WebSphere Commerce Analyzer is now integrated with Tivoli Web Site Analyzer allowing you to create reports based on Tivoli Web Site Analyzer's click stream analysis capabilities.

2.6.14 Security

Various security upgrades have been applied over the releases. For more information please refer to the WebSphere Commerce documentation. However, we would discuss the impact of some of those to encrypted data in this book during migration.

2.6.15 Caching

Server-side caching improves response time and reduces system load thus boosting performance and reducing infrastructure costs. The WebSphere Commerce servlet and page caching is now part of the WebSphere Application Server dynamic caching infrastructure (Dynacache).

Dynacache

WebSphere Application Server offers a built-in dynamic cache service for serving dynamic content and caching data. There is no time-consuming installation and integration work needed to activate. DynaCache can be viewed as a sophisticated hash table, where Java object store and retrieve commands are executed in memory.

Some of the features of WebSphere Application Server dynamic cache (DynaCache) are servlet/JSP result cache, replication support, invalidation support, Edge of Network caching support and tools to assist in configuring the cache and monitoring runtime.

The cache policy is defined by a policy based cachespec.xml file. Some of the cacheable objects defined by this policy include: servlet/JSP files, command beans, Web services, cache and dependency ID's generation rules, replication policy, invalidation rules, time-out, and priority for least recently used eviction algorithm. The cachespec.xml file resides inside the Web Module WEB-INF directory and a copy of this file can be placed in the application server properties directory. It is also flexible and modifiable at runtime.

This type of cache also supports fragment caching. This is the ability to cache portions of a page. For example, instead of caching a full page MyJSP.jsp, you can choose to cache fragments, which can include the header, footer, sidebar and the main area. All of these fragments are dynamic pages that are part of the larger dynamic page for MyJSP.jsp. There is a performance gain in fragment caching but it is not dramatic because the non-cached parts of the page determine the performance.

Replication support enables cache sharing (central cache) and replication (local cache) among multiple servers and tiers. Replication uses a built-in high performance JMS broker messaging system as its underlying engine for data replication. In local cache replication among server, if there is a update to the cache, the cache server publishes a message regarding the change to the messaging broker and the changes are automatically replicated to the other servers.

Some factors to consider for fragment caching include reusability, invalidation and variations. Fragments should be able to be reused frequently enough by

multiple users or pages. Fragments should be stable over a long enough period of time for meaningful reuse to occur.

The goal of fragment caching with DynaCache is to maximize fragment reusability and cache utilization by breaking a page into smaller and simpler page fragments. This makes any caching, even a highly dynamic page, possible.

The caching should take place as close to the user as possible, taking into consideration security and personalized data factors. For example, non-user specific data can be placed closest to the user (like on an Edge Server), but secure information should be placed behind a firewall. Pages need to be fragmented out as much as possible so they can be cached at different locations. The order for placing this from close to far is: Edge server, IBM HTTP Server, WebSphere Application Server, Database Server.

WebSphere Commerce has dynamic cache enabled by default, but the cachespec.xml contains no entries for any of the provided business models. Instead, sample subsets of cachespec.xml files are provided to enable caching of the business models. They also support invalidation using different techniques.

WebSphere Commerce comes with a Dynamic Cache Monitor enterprise application that displays statistics and contents of the dynamic cache. The monitor can be accessed via a Web browser accessing the following URL:

`https://<hostname>:8002/cachemonitor`

This application is provided as an installable Enterprise Archive (EAR) file and can be installed in a server that has dynamic cache enabled. With this application, you can monitor caching interactively and invalidate pages.

In summary, almost everything can be cached if the pages are broken down into smaller fragments. DynaCache can be viewed as a sophisticated hash table that can increase performance and reduce infrastructure costs. Among its many features include the ability to off load caching from memory to disk, full page and fragment caching, invalidation support (either manual using cache monitor, time-out based, event driven, WAS API, or invalidation command) and replication across servers and Edge of Network caching support. DynaCache is configured through a policy-based XML file called cachespec.xml. Some factors to consider on deciding what to cache include reusability, invalidation and variations of the JSP fragments. Finally, for best performance, caching should be placed as close to the user as possible.

2.6.16 Product Advisor

The Product Advisor creates an interactive online product catalog that provides customers with different ways of finding what they want, using shopping metaphors. WebSphere Commerce 5.4 introduced Product Advisor Administration user interface, which included four metaphors to help customer compare products, explore products and provide guidance in making appropriate selection.

3



Migration Strategy and Planning

This chapter provides an overview of the migration process and discusses the initial stages of planning that are involved in choosing the appropriate migration approach. This chapter also includes the skills needed to complete migration. Hardware and software requirements are also defined.

This chapter is organized in the following sections:

- ▶ Migration Strategy considerations
- ▶ Migration Planning
- ▶ Product versions mapping
- ▶ Migration approach for development environment
- ▶ Migration approaches for runtime environment
- ▶ Approach used for this migration

3.1 Migration Strategy considerations

Migration to WebSphere Commerce 5.6 is a complex undertaking. Regardless of the migration approach you choose, it will require a significant investment in time and resources. It will often be cost effective to include any pending site redesign or feature addition projects along with your migration effort. Here are some of the factors to consider.

3.1.1 Add or replace functionality

Consider adding new features and functionality that are available in WebSphere Commerce V5.6. Many of the marketing, advertising and convenience features built into the new version provide mechanisms to increase revenue and improve the customer experience.

Additionally, if you have custom implementations that provide similar functionality to those new built-in features, you may want to consider the advantages of using the “stock” implementations provided in the product.

By rebuilding your store based on one of the new “Starter Store” models you can take advantage of the new functionality rather than implementing or maintaining these features in your existing site. Often you can reduce the amount of customization you have to manage which translates directly into lower operating costs.

3.1.2 Take advantage of DynaCache

Redesigning JSPs to utilize the DynaCache page and fragment caching technology may result in measurable improvements in throughput by reducing the load on the application server and database. If you employ personalization, isolating the personalized content within JSP fragments is recommended to ensure the efficient operation of the site. Consider using the new “Starter Stores” as a model.

3.1.3 Optimization

Customers that are satisfied with their existing set of features and functionality and are interested in moving their existing implementation to the current version without any changes may still benefit from a site redesign if the load on the application server and database components can be reduced.

3.1.4 External product and user management

If you manage your persistent data externally, then there may be little benefit derived from performing a migration. Some examples may include:

- ▶ Third-party or custom catalog management systems
- ▶ LDAP authentication
- ▶ Order status and history retrieved on-demand from an external source

3.1.5 How does custom code impact the process?

Regardless of the approach chosen for the migration, you will have to recreate and test your custom code to account for changes in the new version.

WebSphere Commerce V5.6 introduces the concept of a “Master Catalog”. The Master Catalog must be a true tree with no loops in the structure. Items must now have parents and the parent products can only exist in a single category in the Master Catalog. If you manage your product information in an external system, consider how these new requirements will impact your integration code.

Several changes in the structure and deployment of custom code will require your attention.

- ▶ String Converters in Access Beans are no longer supported
- ▶ Security Roles are now required for Enterprise Java Beans
- ▶ Fine Grained Access Control methods have changed
- ▶ Command parameter validation methods have changed
- ▶ User registration and logon mechanisms have changed
- ▶ Calculation framework customization is now policy based
- ▶ Access to OrderItems through the OrderItemAccessBean has been changed to use a new Item Bean instead

Note: For further details refer to the product information center at the following URL:

<http://publib.boulder.ibm.com/infocenter/wc56help/index.jsp>

3.1.6 Prepare a detailed Plan

Review with your development team the requirements and then prepare a detailed migration plan that includes the refactoring and testing effort required to implement these changes on your site. The following section cover Planning more in detail.

3.2 Migration Planning

WebSphere Commerce integrates with several other software products to provide award winning e-commerce solutions. Migrating WebSphere Commerce can be a complex and challenging task given the number of products and subsystems involved. A complete migration plan must include all the necessary upgrades to application code and WebSphere Commerce servers, but that plan must also consider business requirements, while ensuring that during the migration, service cannot be negatively impacted in any significant way. That is, business must continue during the migration.

Migrating to a new version of WebSphere Commerce will have an impact on several aspects of your business. WebSphere Commerce is far more than just application code; along with the application code, there are development environments, build processes, runtime environments and other aspects that must be considered.

While planning it is important to consider:

- ▶ The customization level for the application to be migrated
- ▶ Development environment (including source code management)
- ▶ Testing environment
- ▶ Runtime environments
- ▶ Deployment processes
- ▶ Database size
- ▶ Integration level with third-party applications
- ▶ How well the application to be migrated complies with the recommended best practices and standards.
- ▶ Education and training

3.2.1 Skill requirements

People who will be installing and configuring WebSphere Commerce V5.6, WebSphere Commerce development environment, and those involved in the migration process should have knowledge in the following areas:

- ▶ IBM DB2 Universal Database
- ▶ Understanding of relational database concepts
- ▶ Ability to perform basic SQL queries
- ▶ IBM HTTP Server administration skills
- ▶ Understanding of TCP/IP, HTTP and SSL protocols

3.2.2 Hardware and software prerequisites

The following sections represent the minimum requirements for hardware and software involved in the installation and configuration of both production and development environments.

Important: After installing all the required software you must install WebSphere Commerce V5.6 Fixpack 1 required for migration. It can be downloaded from the following URL:

<http://www-306.ibm.com/software/genservers/commerce/wcpe/support/>

Production Environment

Before installing WebSphere Commerce ensure your system meets the following hardware and software prerequisites.

Hardware prerequisites

Intel™ Pentium® III 733 MHz (1 GHz or higher recommended for a production environment) IBM-compatible personal computer. This machine should not be running any other applications other than those required by WebSphere Commerce and the system must have the following:

- ▶ Memory:
 - A minimum of 1 GB of RAM per processor for the first WebSphere Commerce instance with optional WebSphere Commerce Payments instance.
 - Each additional WebSphere Commerce instance with optional WebSphere Commerce Payments instance will require an additional 512 MB of RAM per processor.
- ▶ Disk space:
 - A minimum of 2 GB of free disk space on your target installation drive.
 - Additional 900 MB temporary disk space in the location defined by the Windows %tmp% environment variable.
 - If your machine is formatted with FAT partitioning and the partition is over 1.024 GB, you will need twice as much free disk space. If your machine is formatted with FAT partitioning and the partition is over 2.049 GB, you will need three times as much free disk space.
 - The paging file size should be double the size of the RAM. For example, 512 MB RAM should have a 1024 MB paging file.
- ▶ Accessories:

- A mouse or other pointing device. (optional)
- A graphics-capable monitor with a color depth of at least 256 colors.
- A CD-ROM drive
- ▶ Connectivity: A local area network (LAN) adapter that is supported by the TCP/IP protocol.

Software prerequisites

Ensure that any system on which you plan to run the WebSphere Commerce V5.6 installation wizard meets the following minimum software requirements:

- ▶ Ensure that you have one of the following Microsoft Windows operating systems installed:
 - Microsoft Windows 2000, Server Edition with service pack 4 (or higher)
 - Microsoft Windows 2000, Advanced Server Edition with service pack 4 (or higher)
 - Microsoft Windows Server 2003, Enterprise Edition
 - Microsoft Windows Server 2003, Standard Edition

Note: Microsoft Windows Server 2003 is not supported for the WebSphere Commerce development environment.

In addition to the required service pack levels, you should ensure that your system has the latest critical fixes installed.

For more information about service packs and critical fixes, refer to the Microsoft Windows Update site:

<http://www.windowsupdate.com>

- ▶ Ensure that the system is DNS enabled so that there is a host name and domain present. Pure IP address environments are not supported by WebSphere Commerce.
- ▶ Ensure that you disable any virus scanning software active on the system. Virus scanning software often interferes with the installation by causing problems when changing CDs during the installation.

You can re-enable the virus scanning software immediately after completing the installation.

Development Environment

Before installing WebSphere Commerce Developer, ensure your machine meets the following hardware and software prerequisites.

Hardware prerequisites

Ensure that the system on which you plan to install WebSphere Commerce Developer meets the following hardware requirements as outlined in the following table:

Table 3-1 Hardware prerequisites for development environment

	Minimum hardware	Recommended for lightweight test environment	Recommended for full test environment
Processor	Intel™ Pentium® II processor	Intel Pentium III 500 MHz processor or higher	Intel Pentium III 500 MHz processor or higher
Memory	512 MB RAM	768 MB RAM	1 GB RAM
Hard Disk	4.0 GB hard drive space broken down as follows: WebSphere Studio Application Developer 2.1 GB WebSphere Commerce toolkit 1.9 GB	4.0 GB hard drive space broken down as follows: WebSphere Studio Application Developer 2.1 GB WebSphere Commerce toolkit 1.9 GB	4.0 GB hard drive space broken down as follows: WebSphere Studio Application Developer 2.1 GB WebSphere Commerce toolkit 1.9 GB
Display resolution	800 x 600 display resolution	1024 x 768 display resolution	1024 x 768 display resolution

Software prerequisites

Ensure that the system on which you want to install the WebSphere Commerce Developer Environment meets the following software requirements.

- ▶ Ensure that you have one of the following Microsoft Windows operating systems installed:
 - Microsoft Windows 2000, Professional Edition with service pack 3 (or higher)
 - Microsoft Windows 2000, Server Edition with service pack 3 (or higher)
 - Microsoft Windows 2000, Advanced Server Edition with service pack 3 (or higher) Product versions mapping

Note: Microsoft Windows Server 2003 is not supported for the WebSphere Commerce development environment.

3.3 Product versions mapping

This section illustrates the product versions mapping we used in our scenario while migrating from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

Table 3-2 illustrates the application environment, products and versions before and after the migration.

Table 3-2 Product versions mapping, production environment

Pre-Migration	Post-Migration
WebSphere Commerce Suite V5.1 Start	WebSphere Commerce V5.6 Professional
WebSphere Application Server V3.5.6	WebSphere Application Server V5.0.2
DB2 Universal Database V7.2.5	DB2 Universal Database V8.1.5
IBM HTTP Server 1.3.12	IBM HTTP Server 1.3.26
IBM Payment Manager 2.2.1.0	IBM WebSphere Commerce Payments is a base (integrated) component within WebSphere Commerce V5.6

Table 3-3 illustrates the development environment, products and versions before and after the migration.

Table 3-3 Product versions mapping, development environment

Pre-migration	Post-Migration
Visual Age For Java 3.5	WebSphere Studio Application Developer V5.1.1
WebSphere Studio 3.5	WebSphere Commerce V5.6 Toolkit

3.4 Migration approach for development environment

Only one migration approach for the Development environment is supported - In-place migration. You must install WebSphere Commerce V5.6 development environment on the same machine that hosts the old development environment - WebSphere Commerce Studio V5.1.

After migration only the new development environment could operate on this machine and as such it is important that you have another WebSphere Commerce Studio V5.1 environment if you need to continue to develop or maintain WebSphere Commerce Suite V5.1 site.

The following is a general ordered list of tasks that are required to complete development migration:

1. Backup: Backup files, database and environment assets
2. Install/migrate software stack
3. Perform pre-migration steps
4. Import files to WebSphere Commerce V5.6 development environment
5. Generate deployment code and access beans
6. Apply changes for code migration
7. Migrate WebSphere Commerce development instance
8. Migrate database schema and unencrypted data
9. Migrate encrypted data
10. Setting up Test environment and Testing the migrated application

3.5 Migration approaches for runtime environment

This section describes three different approaches to migrating from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6 runtime environment. A number of factors goes into deciding which approach is best suited to the system to be migrated such as: existing configuration, hardware resources available, and acceptable downtime of the production system during course of migration.

When the appropriate migration approach is chosen, the downtime of the production system can be kept to a minimum. The following sections describes options provided by the migration tools of WebSphere Commerce V5.6. Consider the approaches and decide which is best suited for your system. The approaches are as follows:

- ▶ Switch-Over migration scenario- involves creating a separated WebSphere Commerce V5.6 environment on a remote node while keeping WebSphere Commerce Suite V5.1 environment operational.
- ▶ Co-existence migration scenario - involves migrating to WebSphere Commerce V5.6 environment and creating a new instance while the WebSphere Commerce Suite V5.1 environment is still running on the same machine. This is possible if the WebSphere Commerce Suite V5.1 node has

sufficient capacity to operate both of the instances (and software stacks, for single node install) at the same time.

Note: This approach is not supported for Microsoft Windows platforms.

- In-place migration scenario- migration is done on one system (but a test server is highly recommended).

3.5.1 Switch-Over migration scenario

This approach is suitable when the necessary hardware to duplicate the current runtime environment is available to create a test environment where the migration takes place. Once this test environment is fully migrated and tested this test environment becomes the new production environment and the old production environment is taken off line.

With this approach the server downtime is kept to a minimum. The migration is carried out and tested on the test environment while the production servers remain running in parallel. By the time the migration and testing are complete, the database from the migrated environment will be out of sync with the production database. To reconcile the data, the production environment (still on WebSphere Commerce Suite V5.1) must be taken off-line and the database re-migrated based on the already learnt lessons. Once the migration is complete the test environment will be switched-over as the production environment.

Turning the newly migrated environment into production is a relatively simple task to do. Figure 3-1 depicts a high level layout of the Switch-Over migration scenario approach implementation. Once the test environment has been fully tested and the database reconciled, at the firewall or router level, depending on the configuration used, the NAT (Network Address Translation) rules are changed to route all the user traffic to the new runtime. The original WebSphere Commerce Suite V5.1 runtime remains active to serve as the contingency environment in the case of any problems after may occur turning the migrated environment into production.

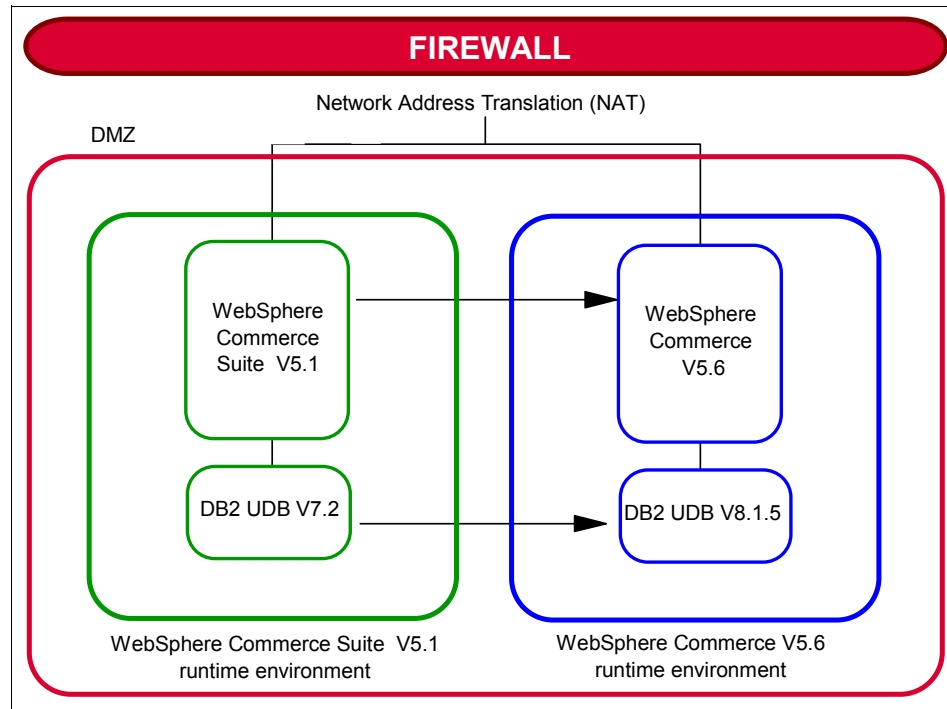


Figure 3-1 Switch-Over migration scenario

The following is a general ordered list of tasks that are required to complete a Switch-Over migration:

- ▶ Complete migration of a development environment, as discussed in the In-place migration section.
- ▶ Complete pre-migration steps required on production server
- ▶ Backup production system
- ▶ Perform pre-migration actions
- ▶ Install WebSphere Commerce V5.6 components on the test server
- ▶ Migrate instance
- ▶ Backup database from the production server
- ▶ Restore database to the test server
- ▶ Run scripts to migrate database
- ▶ Deploy migrated code from deployment machine
- ▶ Complete required migration tasks on the test server
- ▶ Test migrated instance
- ▶ Take the WebSphere Commerce Suite V5.1 production server offline
- ▶ Backup database from the production server
- ▶ Restore database to the test server
- ▶ Run scripts to migrate database

- Bring WebSphere Commerce V5.6 test server online to become the new production server

We choose and document this approach in this book, to minimize both the production downtime and the risk involved in migration. Section 3.6, “Approach used for this migration” on page 49 describes more in detail the migration path we followed.

3.5.2 Co-existence migration scenario

This approach is suitable if your WebSphere Commerce Suite V5.1 server has sufficient capacity to operate both instances WebSphere Commerce Suite V5.1 and WebSphere Commerce V5.6 environments.

It involves setting up a WebSphere Commerce V5.6 instance on WebSphere Commerce Suite V5.1 machine while the original instance is still operational and then switching the instances from old to new when the environment is ready.

Note: This migration approach is not supported on Microsoft Windows platforms. Additional information about this migration approach can be found in the product guide.

3.5.3 In-place migration scenario

In-place migration is performed on the production server. This approach typically has the longest server downtime in comparison to the other two approaches, although not necessarily the entire migration has to be done all at once. Migrating individually some components may help to minimize the production server downtime.

This approach should be considered when there are limited hardware resources and only the production server is the available system for migration (i.e. single-node implementations). This is assuming that neither high availability is required nor additional systems are available. If this is the case then this approach offers an option to migrate to WebSphere Commerce V5.6.

Having a multiple-node environment may provide the chance to consolidate functionality and nodes, freeing up some nodes should allow you to migrate them off line. The cost of doing this would be a reduction in the processing capacity but the site would still remain operational.

For this particular approach a very detailed and careful planning is required. Although supported, this approach may not be preferred if server downtime and working directly on the production server is a concern.

The following is a general ordered list of tasks that are required to complete an In-place migration:

- ▶ Take WebSphere Commerce Suite V5.1 production server offline
- ▶ Backup production system
- ▶ Perform pre-migration actions
- ▶ Install (for coexistence) or update software levels to WebSphere Commerce V5.6
- ▶ Perform instance migration
- ▶ Migrate database
- ▶ Deploy migrated code from the development machine
- ▶ Complete required migration tasks on the test server
- ▶ Test migrated instance
- ▶ Bring WebSphere Commerce V5.6 test server online to become the new production server

3.6 Approach used for this migration

In our approach our main goal was to minimize the downtime of the production server. To achieve this we chosen the Switch-Over migration.

Our final production environment consisted of a three nodes - a WebSphere Commerce node with a remote database node and a remote WebSphere Commerce Payments node. Initially, however, we migrated the WebSphere Commerce and database to a single node. Then, once we had successfully tested the migrated environment and we were satisfied with migration to WebSphere Commerce V5.6, we moved the database to a remote database server creating a three node environment as the original system topology.

While we were migrating the solution, the first stage of the migration, the WebSphere Commerce Suite V5.1 remained running thus we had minimal downtime. As soon as we had migrated we switched over the actual production environment from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6, WebSphere Commerce Suite V5.1 remained operational as our backup incase of problems.

Below are described the high level tasks we performed to do a complete migration on the development and runtime. It is important to migrate the development environment first so that you have the migrated code for deployment after you have migrated the production environment:

1. Migrate development environment
2. Migrate production environment

3. Deploy the code changes from the development environment to the production environment to finish the migration of the production environment

3.6.1 Detailed overview of development migration steps

Since all the following steps are performed on the development environment there should not be any downtime on the production environment.

1. Backup: Backup files, database and environment assets
2. Install/migrate software stack:
 - upgrade Database level to DB2 Universal Database V8.1.5
 - Install WebSphere Commerce Development environment on the same system as your WebSphere Commerce Suite V5.1 development environment. Switch-Over migration scenario for this environment is not supported. This involves installing: WebSphere Studio Application Developer, eFixes and WebSphere Commerce V5.6 Toolkit
 - Reconfigure development environment to use DB2 database by running the “setdbtype” script
 - If required, install VisualAge for Java V4.0 on a separate machine, import repository and apply fixes
3. Pre-migration actions
 - Setup custom constraints
 - Before migrating the database you need to drop all your custom constraints
 - Once you have finished migrating the database, the custom constraints you dropped need to be re-established.
 - Ensure that your data meets various other premigration requirements. This includes messages, messages types, payments, catalog, members and access control data.
4. Prepare code for importation into the WebSphere Commerce V5.6 development environment
 - Export Java project from WebSphere Commerce Studio V5.1 as JAR
 - Export EJB projects from VisualAge for Java V3.5.3 and import into VisualAge for Java V4.0 to migrate the EJB level from 1.0 to 1.1. Export EJB project from VisualAge for Java V4.0 as EJB 1.1 JARs.

You could also recreate EJBs in the WebSphere Commerce V5.6 development environment instead of installing VisualAge for Java V4.0 and channeling through it.
5. Import files to WebSphere Commerce V5.6 development environment
 - Import EJB project into WebSphere Commerce V5.6 development environment in WebSphereCommerceServerExtensionData

- Import Java project into WebSphere Commerce V5.6 development environment in WebSphereCommerceServerExtensionLogic
6. Generate deployment code
 - For full test environment: From the EJB JAR pop-up menu, click Generate > Deployment and RMIC Code
 - For lightweight test environment: From the EJB JAR pop-up menu, click Generate > Test Deployment Code
 7. Generate Access beans
 8. Code Changes
 - Fix any WSAD compilation errors
 9. Migrate WebSphere Commerce Suite V5.1 development environment instance: You would need to copy the product.xml and product.dtd file to the <WCS51_home> /xml directory and update: edition, release, fixpack and path before you migrate instance
 10. Migrate database schema and unencrypted data:

First run the database migration script in -precheck mode to find any inconsistencies which could be discovered with actually migrating the data.

This does not, obviously, guarantee the discovery of all the issues you may find while doing the actual database migration. As such you may want to position your first attempt at database migration to discover any additional database inconsistencies. The migration script does not interrupt when it finds any issues and this could allow the first migration run to act as a more thorough check for database consistency for migration. After the first test-run, you can restore the database and fix the problems discovered and re-run the migration scripts. You may want to tune database migration performance if it took a long time during the first run. Please refer to the behavior of migration scripts found in Appendix C, “Migration scripts” on page 255.

In order to ease the task of migrating the database second time over we created custom scripts to perform preparation and migration tasks. These scripts can be found in Appendix C, “Migration scripts” on page 255.
 11. Migrate encrypted data: At this step you migrate user password data and which you could have chose to encrypted viz, credit card data.
 12. Setting up Test environment and testing the migrated environment

3.6.2 Detailed overview of production environment migration

In our scenario since we performed Switch-Over migration we installed and configured WebSphere Commerce V5.6 environment on another machine, say

environment #2, and restored a backup of production database on the environment #2. When we were satisfied that the migration had gone well:

- we took our environment #2 and split it into a multiple-node environment and re-tested to ensure everything was operational
- we took our production server environment, say environment #1, offline
- restored the latest copy of our database from environment #1 to environment #2,
- remigrated the database
- switched over environment #2 as the new production environment

Following this approach we were able to minimize our production downtime. Here are the detailed summary of steps we undertook to migrate to environment #2 before splitting it into a multiple-node environment:

1. Decide migration approach

Based on the differences of migration approaches we decided to go with Switch-Over migration to minimize the production server downtime.

2. Recreate production environment on environment #2, as discussed above.
3. Premigration actions: Same actions as on the development environment. You may, however, discover some additional issues on this database due to additional data.
4. Bring system offline
5. Install/upgrade system
6. Migrate instance. Before executing instance migration script, ensure that:
 - the product.xml and product.dtd have been copied to the appropriate places and the content updated as required
 - WebSphere Application Server security is turned off
 - if you are using IBM HTTP Server then you need to uncomment some lines in the httpd.conf file
 - the default ports 8000, 8002, 8004 are available for WebSphere Commerce V5.6 tools usage, or change the defaults
 - Execute instance migration script
 - Re-enable WebSphere Application Server security, if desired
 - Configure Web Server
7. Deploy your custom code
8. Performance tune your database: Sample script can be found in the bin directory of WebSphere Commerce V5.6 install directory:

updateDB2Configuration.sh database_name log_name.

The database tuning parameters given in this script should be considered as as the minimum required. It is highly recommended that you check the values of this script and evaluate your specific database needs, e.g. you may already be using higher values and, in such a case, you should not execute this script.

9. Migrate database schema and unencrypted data: Same actions as performed on the development environment
10. Migrate encrypted data
11. Post migration actions: Apply changes unique to your scenario, e.g. optional cache setup, etc.
12. Migrate WebSphere Commerce Payments
13. Migrate Staging server: We did not have staging server in our environment
14. Migrate member subsystem if you have setup with LDAP: We did not have a separate member subsystem component in our scenario
15. Start components: You may want to run test case to ensure that the site is up and running to your needs.

4

Commerce Application used during the migration

This chapter provides an overview of the application used during the WebSphere Commerce migration. An overview of the application is given along with a description of the customizations that were made to the default architecture.

The chapter is organized into the following sections:

- ▶ Commerce application
- ▶ Commerce customization
- ▶ Catalog
- ▶ Database

4.1 Commerce application

CarrotInk.com is an Internet-based retailer of inkjet printer cartridges and supplies. CarrotInk.com sells after-market and OEM brand printer supplies. The company supplies as many as 400 individual cartridge types, supporting thousands of different printer models. Before migrating to WebSphere Commerce Suite V5.1, the original site contained more than 800 static HTML pages. The pages were managed manually and the need to upgrade to a new system became prevalent.

Carrot Ink needed a solution that would provide easy maintenance of the product catalog and Web site design. Another issue was that customers could not register with the site and store billing and shipping information, therefore no order history was provided to the customer.

The solution was to migrate the existing site to WebSphere Commerce Suite V5.1. The products catalog is now easier to update, the site look and feel is easier to update, and the performance of the site is excellent. Back-end integration has been facilitated, helping Carrot Ink to process and fulfill more orders.

This section gives an overview of the application environment of the commerce application.

4.1.1 Application environment

The application environment has been implemented as two node which consists of an WebSphere Commerce Suite V5.1 application node and a database server node. These servers are mirrored servers running Raid 1. Both of these servers are behind a firewall.

Hardware Configuration

The following is a list with the hardware configuration of the application and database servers:

- ▶ Microsoft Windows 2000 Server, Service Pack 4
- ▶ Pentium III, 1266 MHz
- ▶ 2 GB Ram
- ▶ 40 GB Hard Disk

Application Server Tier

The following shows the software and levels installed and configured on the application server tier:

- ▶ WebSphere Application Server V3.5.6

- ▶ WebSphere Commerce Suite V5.1 Start Edition
- ▶ IBM HTTP Server V1.3.12

Database Server Tier

For the database node, DB2 Universal Database V7.2.5 is installed and configured.

4.2 Commerce customization

Customization of WebSphere Commerce related to the application has been minimal. The site design was based on the InFashion sample SAR file. The application's customization can be defined in three areas: shopping flow, overriding of default controller commands, and database tables. The following sections describe these customization in more detail.

4.2.1 Shopping flow

The default functionality of the shopping flow has been modified to provide custom account creation, dynamic shiprates, expected delivery dates, payment processing, offline orders, and checkout page flow.

Account Creation

First-time customers are allowed to create an account as they navigate through the shopping flow. After adding an item to the shopping cart and continuing with the shopping process by leaving the shopping cart, a page is displayed that allows registered users to logon or unregistered users to create an account. The registered user logs on with their email address and password and continues with the checkout process. The unregistered user fills in their email address but specifies they are a new user. They continue through the checkout process by entering their shipping and billing information. After the billing information is completed, the next page prompts the user to enter a password to be used with their account. After entering a password, the account creation process is complete and the user is forwarded to the final step of the checkout process.

The default account creation process from InFashion has also been implemented.

Dynamic Shiprates/Expected Delivery Dates

The shipping select page has been customized to implement dynamic shiprates and expected delivery dates. The dynamic shiprates are based on factors such as weight of the order and the zip code the order will be shipped. The rates are based on each shipping provider's rate system.

Expected delivery dates have also been implemented for the shipping providers. Custom code was written to calculate the number of expected days that it will take to receive the order. The factors for calculating expected delivery dates are based on the shipping policies of Carrot Ink and each shipping provider. These include exclusion of holidays, non-working days such as Saturday and Sunday, and shipping cut-off times.

Payment Processing

Payment Manger was not implemented in the original the application. The default controller command used for Payment Manager was overridden to implement custom payment processing. The application implements an authorization company for credit card verification. Once the credit card has been verified the order is processed and manually authorized. However, for the purpose of this book, the migration will include the implementation of Payment Manager for order processing.

Offline Orders

Offline orders have also been implemented. At the checkout page, customers have the option of placing the order offline. The customer is given a form to complete which becomes a mail order. Once the customer service department receives the mail order, the order is processed. No order is created in WebSphere Commerce for offline orders so no custom controller commands were written to handle the offline order process. A simple view command was created to display the offline form once the offline order has been submitted.

Billing/Shipping Selection

Another simple modification was to change the order of the select billing address and select shipping address pages. After leaving the shopping cart, the customer is directed to the select shipping address page, followed by the select billing address page.

The application model flow was based upon the InFashion sar file. Figure 4-1 illustrates the default shopping flow as packaged with the InFashion store.

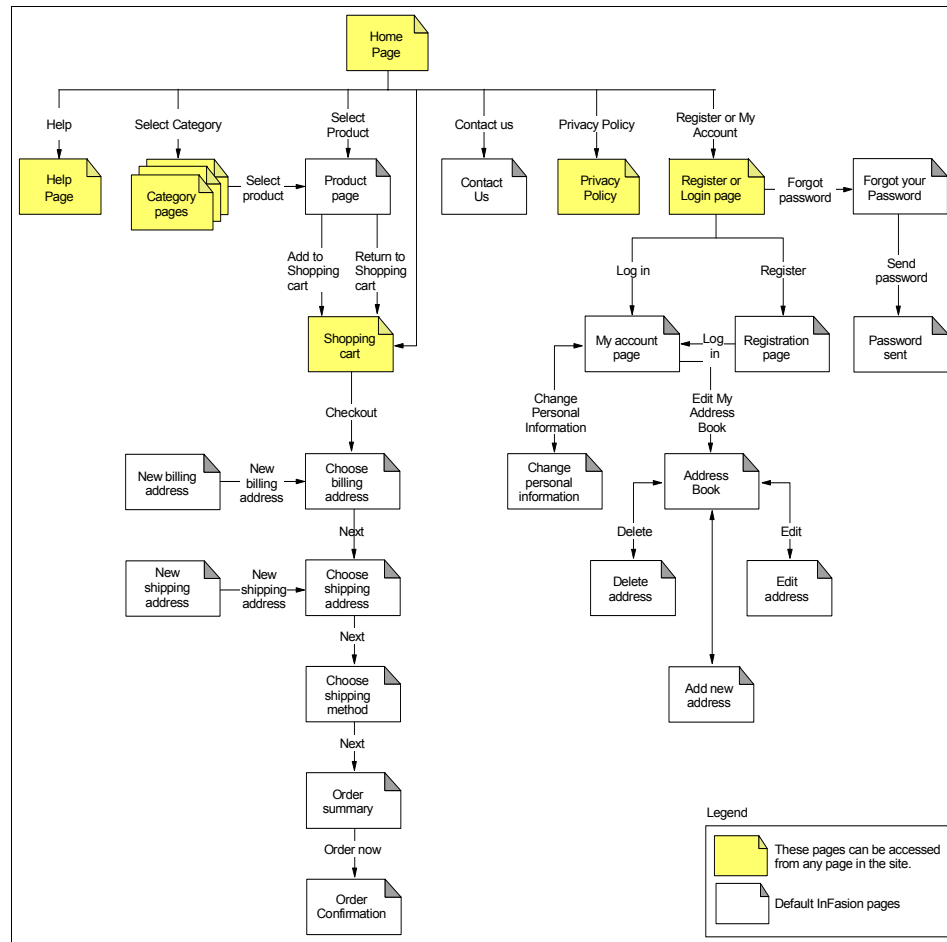


Figure 4-1 InFashion default shopping flow

Figure 4-2 illustrates the customized shopping flow of the application. The changes from the InFashion flow are shown in the shaded boxes.

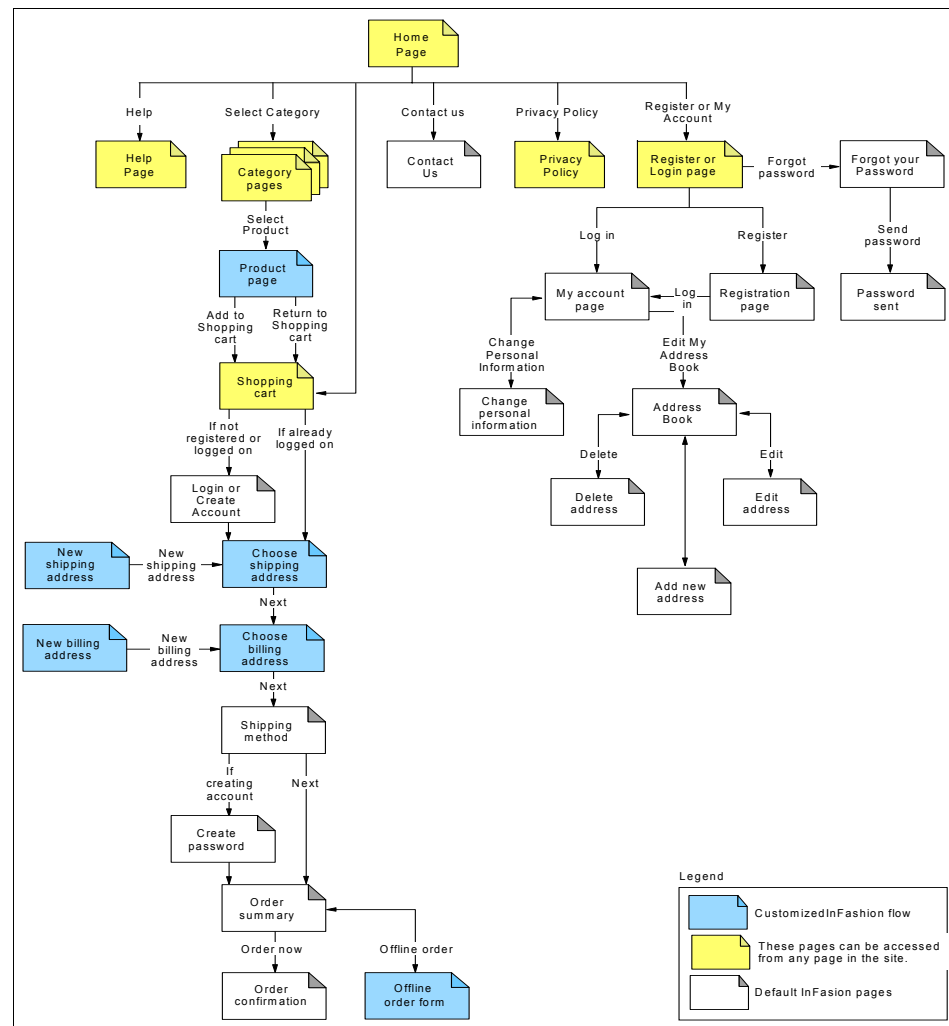


Figure 4-2 Customized shopping flow

4.2.2 Commands

Only a few commands from the default architecture were overridden in the application. The custom commands are used in areas of category display, dynamic shipping rates, and payment processing. Commands were also used to send customers their password and increase the functionality of the search.

Table 4-1 provides a summary of the custom commands used with the application.

Table 4-1 Custom Commands

Default Command(s)	Override Command	Description
ApplyOrderShippingChargesCmdImpl.java	ApplyOrderShippingChargesCmdImpl.java	Implements dynamic shiprates for each shipping provider. The shiprates on factors such as weight of the order and to what zipcode the order will be shipped.
CategoryDisplayCmdImpl	CategoryDisplayCmdImpl	Verifies that the category referenced is valid and exists in the catalog. If category is valid, the category is displayed. This command is used primarily for customers who have bookmarked a category page but the category reference no longer exists in the database.
DoPaymentVerifyCmdImpl	DoPaymentCmdImpl DoPaymentMPCCmdImpl	Bypasses Payment Manager. Authorizes credit card at point of order processing. A request is sent to authorization company and credit card is verified.
SendPasswordCmd SendPasswordCmdImpl	N/A	Accesses the database and retrieves customer logon id and password. A message is then sent to the customer with their account information.

Default Command(s)	Override Command	Description
TXSearchSampleCmd TXSearchSampleCmdImpl	N/A	Sample commands provided by IBM for use with the DB2 Text Extender. The controller command generates a SQL statement that is then used to query the database to return search results.

4.2.3 Database tables

Custom tables were added to increase functionality. Tables were added to implement dynamic shiprates. Other tables were added to store potential fraudulent payment transactions. Table 4-2 provides a summary of custom tables that were created for use with the application.

Table 4-2 Custom Tables

Table	Description
RURALZIPCODES SHIPRATE SURCHARGE ZIPCODE ZONE	Implements dynamic shiprates for FedEx, UPS, USPS. Based on factors such as zipcode, shipping zone, and order weight, the application will deliver real time ship rates to the customer.
XBLOCKIPS XPOTIPS	Non-Commerce related tables. Part of external application that records potential fraudulent payment transactions.

4.3 Catalog

The product catalog includes all the items sold on Carrotink.com. The catalog contains the hierarchy for categories with the items being associated with the bottom level category. The top categories are the printer brand. The subcategories are the printer models. Some items have a 1:1 relation with their parent category while other items have a 1:M (1 to many) relationship. This 1:M relationship is due to the fact that one ink jet cartridge can work with different printers.

4.4 Database

The database architecture has been modified slightly from the default. No customizations have been made to the WebSphere Commerce schema. However, as mentioned earlier, some tables were added for increasing functionality. Tables were added to implement dynamic ship rates. These tables store data related to shipping zones and ship-to zip codes. Also, an external application was written to record possible fraudulent payment transactions. Two tables were created to store this information. These custom tables are mentioned in Table 4-2.



Part 2

Development environment



Installing WebSphere Commerce Development Environment

This chapter details the steps that should be taken to install the development environment for WebSphere Commerce V5.6. Once complete, you will have a fully working WebSphere Commerce development environment, using DB2 Universal Database V8.1 as the database. All steps in this chapter must be completed before attempting the development environment migration in the next chapter.

This chapter is organized in the following sections:

- ▶ Development environment overview
- ▶ Pre-installation requirements
- ▶ Installing WebSphere Studio Application Developer
- ▶ Installing WebSphere Commerce V5.6 Toolkit
- ▶ Installing DB2 Universal Database V8.1
- ▶ Configuring the development environment for DB2
- ▶ Installing VisualAge for Java V4.0

5.1 Development environment overview

This section describes how we installed the WebSphere Commerce Developer software package. The development environment consists of the following products:

- ▶ WebSphere Studio Application Developer V5.1.1
- ▶ WebSphere Commerce V5.6 Toolkit
- ▶ DB2 Universal Database V8.1.5

In the WebSphere Commerce V5.6 Toolkit, two test environments are available, a lightweight environment and a full environment. During the installation documented in this chapter, both environments will be installed and configured for use with DB2 Universal Database V8.1.5.

Unlike the VisualAge for Java development environment for WebSphere Commerce Suite V5.1, a WebSphere Commerce run-time environment is not required. The servers configured in WebSphere Studio Application Developer are all that are required.

Important: For the purposes of migrating from the WebSphere Commerce Suite V5.1 development environment it is mandatory that you install WebSphere Studio Application Developer V5.1.1 on the same system as VisualAge for Java V3.5.3.

After installation of WebSphere Studio Application Developer V5.1.1 and WebSphere Commerce V5.6 Toolkit, your existing VisualAge for Java will still be intact. However, due to the fact that you must upgrade your database to DB2 Universal Database V8.1, you may no longer be able to use the WebSphere Test Environment in VisualAge for Java.

Important: After installing all the required software you must install WebSphere Commerce V5.6 Fixpack 1. It can be downloaded from the following URL:

<http://www-306.ibm.com/software/genservers/commerce/wcpe/support/>

5.2 Pre-installation requirements

Before installing WebSphere Studio Application Developer V5.1.1 and WebSphere Commerce V5.6 Toolkit, ensure that your machine meets the hardware and software prerequisites found in 3.2.2, “Hardware and software prerequisites” on page 41.

The steps detailed in this chapter are provided under the assumption that you have a fully working WebSphere Commerce Suite V5.1 test environment in VisualAge for Java V3.5.3. It is also assumed that you are using DB2 Universal Database V7.2.5 with a local development database.

5.2.1 VisualAge for Java V4.0 prerequisites

In order to install and configure VisualAge for Java V4.0 for EJB migration, you will need the following components:

- ▶ VisualAge for Java V4.0 installation CD
- ▶ The following fixes from the WebSphere Commerce Studio V5.4 installation CD:
 - EJB-1.1-DeployedTool.zip
 - ivjfix35.zip
 - PQ50159.jar
 - readonly.zip

Note: You will need to contact IBM support to obtain these files.

- ▶ The WebSphere Commerce Suite V5.1 repository file, shipped on the WebSphere Commerce Studio V5.1 installation CD.

5.3 Installing WebSphere Studio Application Developer

This section details the steps that must be taken to install WebSphere Studio Application Developer V5.1.1. Once this section is completed, you will have a WebSphere Application Server test environment configured at the correct fixpack level required to install the WebSphere Commerce V5.6 Toolkit.

Note: DB2 (or Oracle) is required when migrating from a previous version of the WebSphere Commerce development environment. If you are starting development from scratch, you should follow the recommendations in *WebSphere Commerce Developer V5.6 Installation Guide for Windows*.

5.3.1 Pre-installation steps

The following list describes the steps that are needed to be carried out prior to the installation of WebSphere Studio Application Developer V5.1.1:

- ▶ Ensure that the user ID that you are using does not contain double-byte characters.
- ▶ Ensure that the environment variable TEMP or TMP is pointing to a valid temporary directory
- ▶ Ensure that the drive containing the temporary directory (mentioned above) has more than 2GB available.

5.3.2 Installing WebSphere Studio Application Developer V5.1.1

The steps below describe how to install WebSphere Studio Application Developer V5.1.1.

Attention: Remember that you must install WebSphere Studio Application Developer V5.1.1 on the same system as your existing VisualAge for Java environment for WebSphere Commerce Suite V5.1.

1. Run **launchpad.exe** from the root of the first installation CD
2. On the **WebSphere Studio Installation Launchpad** window click the button next to **Install IBM WebSphere Studio Application Developer**.
3. Wait for the **WebSphere Studio Installation Welcome** page to open, then click **Next** to continue the installation. The welcome page may take several minutes to appear.
4. Read the license agreement, select **I accept the terms in the license agreement** and click **Next**.
5. Enter the desired target directory in the **Directory Name** field and click **Next**. We recommend that you do not install into a directory containing spaces in the name.

In our example, we entered the following directory:

D:\WebSphere\WSAD511

Important: Do not install into a directory with a name that contains double-byte characters or special characters such as a dollar sign. Doing so may cause undesirable results such as class path problems in the WebSphere test environment.

Note: The installation program checks to see if WebSphere Studio Application Developer V5.1 is installed. Since this cannot coexist with WebSphere Studio Application Developer V5.1.1, if WebSphere Studio Application Developer V5.1 is detected, a message appears asking if you want the installation program to uninstall it. WebSphere Studio Application Developer V5.1.1 re-uses the same install directory as specified for WebSphere Studio Application Developer V5.1, to preserve links to third-party plug-ins.

6. In the features window you can select the WebSphere Studio Application Developer V5.1.1 features that you would like to install. Select WebSphere Application Server V5.0.2 Test Environment and deselect any other test environment and click **Next**.

Note: If you are upgrading from WebSphere Studio Application Developer V5.1, all features that were previously installed will be pre-selected as well.

7. Click **Next** to install WebSphere Studio Application Developer V5.1.1.
During the installation you will be prompted to insert WebSphere Studio Application Developer V5.1.1 CD 2. Insert the CD and click **OK** to continue.
Refer to *WebSphere Commerce Developer V5.6 Installation Guide for Windows* if you encounter warnings or errors during the installation.
8. When WebSphere Studio Application Developer V5.1.1 is installed, click **Finish** to close the installation window.
9. Click the button next to Exit in the **WebSphere Studio Installation Launchpad** window to close it.

5.3.3 Apply fixes to the test environment

In order to prepare the test environment for the installation of the WebSphere Commerce V5.6 Toolkit, several fixes must be installed to upgrade the level of the WebSphere Application Server test environment to the same level as is required by the WebSphere Commerce V5.6 run-time environment. These fixes are provided in the WebSphere Application Server fixpack CD.

Check that the following has been completed before applying fixes:

- ▶ WebSphere Studio Application Developer V5.1.1 with WebSphere Application Server V5.0.2 Test Environment is installed on the machine.
- ▶ Any applications running on your machine have been stopped.

The fixpack installation consist of three steps:

1. Prepare for fixpack installation
2. Install WebSphere Application Server V5.0.2 cumulative fix 3
3. Install interim fixes

The installation procedure is described in the following sections.

Prepare for fixpack installation

Before the fixpack can be applied, the WebSphere Test Environment inside WebSphere Studio Application Developer V5.1.1 must be prepared. The following list describes the necessary steps to configure the test environment for the fixpack installation.

1. Insert the WebSphere Application Server Fixes CD from your WebSphere Commerce package provided with WebSphere Commerce Developer into the CD-ROM drive.
2. From a Windows command prompt, change to the directory CD-ROM drive:\WSAD and issue the following command on a single line:

```
<WSAD_home>\runtimes\base_v5\java\bin\java -jar PrePatcher.jar  
<WSAD_home>
```
3. The pre-patcher will ask you to confirm the location of your WebSphere Studio Application Developer V5.1.1 installation directory. Validate the location and enter Y followed by **Enter** to continue.

The last message from the pre-patcher should be:

```
All done... You should be able to apply WebSphere v5.0.2 Cumulative fix to  
your WTE.
```

Now you can continue with the next step of actually applying the fixes.

Install WebSphere Application Server V5.0.2 cumulative fix 3

Apply WebSphere Application Server Version V5.0.2 Cumulative Fix 3 to your WebSphere Studio Application Developer installation as follows:

1. Copy the updateInstaller directory from the root of the WebSphere Application Server Fixes CD to a temporary location on the hard drive.
2. From a Windows command prompt, do the following:
 - a. Change to the updateInstaller directory that you just created on your hard drive.
 - b. Issue the following commands:

```
set JAVA_HOME=<WSAD_home>\runtimes\base_v5\java  
updateWizard
```

This starts the update installation wizard.

You can close the command prompt at this point.

3. The language prompt will appear. Select the desired language for the Update Wizard and click **OK**.
4. The Welcome page appears. Click **Next**.
5. When prompted to select the product to update, check **Specify product information** and enter the following path in the **Installation directory** field:

<WSAD_home>\runtimes\base_v5

Click **Next**.

Note: The Wizard will show the following message on the page where you specify the location:

WebSphere Application Server Family product - Not found

This just means that the wizard could not find an installation of WebSphere Application Server V5.0.2 and can safely be ignored.

6. Select **Install fixpacks** and click **Next**.
7. In the fixpack directory field, enter the following path and click **Next**.

CD-ROM drive:\BASE\fixpack

The Update Wizard will determine which fixes are applicable for the selected path. This may take several minutes.

8. A list of available fixpacks will be shown. Ensure that **was502f_cf3_win** is selected and click **Next**.
9. The Update Wizard will warn that the following fixes will be uninstalled:
 - PQ78374
 - PQ78419

Click **Next**.

The Update Wizard will begin applying the selected fixpack.

When the installation has completed, click **Run Wizard Again**. This will leave the update installation wizard open to apply the required interim fixes in the next step.

Install interim fixes

Apply the WebSphere Application Server interim fixes to your WebSphere Studio Application Developer V5.1.1 installation as follows. The update installation wizard should still be open from the previous step:

1. From the Update Installation Wizard, select **Install fixes** and click **Next**.

Note: The Install fixes is the third option on the window, easily confused with the first option, **Install fixpacks**.

2. In the **Fix directory** field, enter the following path and click **Next**:

CD-ROM drive:\WSAD\fixes

3. From the list of available fixes, check **PQ82074**, leaving all other fixes unchecked, and click **Next**.

Important: Ensure that only PQ82074 is checked. If you apply more than one fix at a time, the installation will fail.

4. The installation summary will show. Click **Next** to start the fix installation.

5. When the installation has completed, click **Run Wizard Again**.

6. Repeat steps 1 to 4 to install **PQ81989_win**.

7. When the installation has completed, click **Run Wizard Again**.

8. Select **Install fixes** and click **Next**.

9. In the Fix directory field, enter the following path and click **Next**:

CD-ROM drive:\BASE\fixes

10. From the list of available fixes, check **PQ85933_Fix**, leaving all other fixes unchecked, and click **Next**.

11. The installation summary will show. Click **Next** to start the fix installation.

12. When the installation has completed, click **Run Wizard Again**.

13. Repeat steps 8 to 11 for the each of the following fixes, one at a time, in the order they are listed:

- PQ83918
- PQ85469

Important: Fixes must be installed in the order they are listed and one at a time because each fix often replace files from the one before it.

If you do not apply these fixes in the correct order, your test environment may not work properly.

14. After the last fix, PQ85469, has been applied, you can click **Finish** to close the Update Installation Wizard.

5.4 Installing WebSphere Commerce V5.6 Toolkit

Assuming you have completed the installation steps in the previous sections of this chapter, you are ready to install the WebSphere Commerce V5.6 Toolkit. Installing the Toolkit will create the WebSphere Commerce components in WebSphere Studio Application Developer and create the necessary server configuration for the test environment.

As mentioned in 5.1, “Development environment overview” on page 68 the WebSphere Commerce development environment is available with both a lightweight and full test environment. By default, the lightweight test environment is configured to use Cloudscape as the test database.

At a later stage, the full test environment will be installed and both test environments will be configured to use DB2 Universal Database, as this is a requirement for migrating the development instance and database. This is documented in 5.6, “Configuring the development environment for DB2” on page 85.

After finished installation and migration, you can switch between the two environments as you development needs require.

Before continuing with the installation of the WebSphere Commerce V5.6 Toolkit you must ensure that the following have been completed:

- ▶ WebSphere Studio Application Developer V5.1.1 with WebSphere Application Server V5.0.2 Test Environment is installed on the machine.
- ▶ WebSphere Application Server V5.0.2 cumulative fix 3 and interim fixes have been applied to the test environment as described in 5.3.3, “Apply fixes to the test environment” on page 71.
- ▶ No applications are running on your system.

5.4.1 Installing WebSphere Commerce V5.6 Toolkit

The steps below describe how to install WebSphere Commerce V5.6 Toolkit.

1. Insert the WebSphere Commerce V5.6 Toolkit CD in the CD-ROM of the WebSphere Studio Application Developer V5.1.1 machine. The WebSphere Commerce V5.6 Toolkit installation wizard should start automatically. If the installation does not start automatically, run **setup.exe** on the root of the WebSphere Commerce V5.6 Toolkit CD.
2. The language prompt will appear. Select your desired language for the installation wizard and click **OK**. For our example, we selected English.
3. The Welcome page will appear. Click **Next**.
4. Read the license agreement, select **I accept the terms in the license agreement** and click **Next**.
5. The destination path page will appear. Enter the directory in which you wish to install the WebSphere Commerce V5.6 Toolkit and click **Next**.

For our example, we entered the following path:

```
D:\WebSphere\WCToolkitPro56
```

Important: Do not enter a path containing spaces, double byte characters or other special characters.

6. The installation summary window will appear. Ensure that you have sufficient hard disk space and click **Next** to start the installation.
7. Click **Finish** when the installation is complete, to close the installation wizard.

Installation verification

To validate that the installation of the WebSphere Commerce V5.6 Toolkit was successful, do the following:

1. Examine the contents of <wctoolkit_home>\logs\setup.log. If **setup.log** is empty or has errors, you can retry the configuration steps by running <wctoolkit_home>\bin\setup.bat from the command line.
2. Check that the install was able to properly detect WebSphere Studio Application Developer V5.1.1 on your system by ensuring that the following file sets the WSAD_HOME environment variable to <WSAD_home> correctly:

```
<wctoolkit_home>\bin\setenv.bat
```

The file should look like the sample file shown in Example 5-1 with your environment specific values substituted.

Example 5-1 Sample setenv.bat file for WebSphere Commerce V5.6 Toolkit

```
set WSAD_HOME=D:\WebSphere\WSAD511
call setshortname RUNTIME_JARS ..\AppServer\classes
call setshortname RUNTIME_CLASSES ..\AppServer\classes
call setshortname J2EE_JAR ..\AppServer\lib\j2ee.jar
call setshortname XERCES_JAR ..\AppServer\lib\xerces.jar
call setshortname XALAN_JAR ..\AppServer\lib\xalan.jar
call setshortname XMLPARSERAPIS_JAR ..\AppServer\lib\j2ee.jar
call setshortname JAVA_HOME ..\AppServer\java
call setshortname PROPERTIES ..\properties
set path=%JAVA_HOME%\bin;%PATH%
call setshortname WSAD_HOME "%WSAD_HOME%"
call setshortname WCTOOLKIT "%~d0%~p0.."
set WORKSPACE_DIR=%WCTOOLKIT%\workspace
set WAS_HOME=%WSAD_HOME%\runtimes\base_v5
call setshortname WCS_HOME ..
call setshortname WCLOGDIR ..\logs
call setshortname WCTEMPDIR ..\temp
```

You now have a fully functioning WebSphere Commerce V5.6 development environment installed. The development environment is set up to use the lightweight WebSphere Commerce test environment with Cloudscape as database system.

In 5.6, “Configuring the development environment for DB2” on page 85, the test environment will be configured to use DB2 Universal Database, but first we must install DB2 Universal Database V8.1.

5.5 Installing DB2 Universal Database V8.1

DB2 Universal Database V8.1 must be installed in your new development environment in order to complete a successful migration. This is because the full WebSphere Commerce test environment is required. This section details the installation steps.

Note: This guide assumes that your WebSphere Commerce Suite V5.1 development environment uses a local installation of DB2. That is, DB2 Universal Database V7.2.5 is installed on the same machine as VisualAge for Java V3.5.3.

5.5.1 Backup development databases

We recommend that you backup all databases in the development environment that cannot be recreated before you install DB2 Universal Database V8.1, as

databases are not interchangeable between these DB2 Universal Database versions.

In this case you will have a DB2 Universal Database V7.2.5 level of your databases, should you later need to change back to DB2 Universal Database V7.2.5.

In our example, we created a backup of the database MALL, as the remaining databases can be recreated if necessary.

5.5.2 Installation requirements

Before continuing, you should ensure that you have a working development environment that is using Cloudscape, as installed in 5.3, “Installing WebSphere Studio Application Developer” on page 69 and 5.4, “Installing WebSphere Commerce V5.6 Toolkit” on page 75.

As you already have a database on your development system in DB2 Universal Database V7.2.5, you must run the **DB2CKMIG** tool provided on the DB2 Universal Database V8.1 installation disk before installing DB2 Universal Database V8.1. This tool will check your existing databases for conditions that would prevent a successful migration.

To run the **DB2CKMIG** tool, complete the following steps:

1. Insert the DB2 Universal Database V8.1.5 CD, supplied with WebSphere Commerce V5.6 in your CD-ROM drive.

Important: WebSphere Commerce V5.6 is shipped with a special version of DB2 Universal Database V8.1, refreshed to fixpack 5 level. Do not confuse the DB2 Universal Database V8.1.5 CD with a standard DB2 Universal Database V8.1 installation CD.

2. In a Windows command prompt, navigate to the following directory on the DB2 Universal Database V8.1.5 CD:

```
<CD_Drive>db2\Windows\utilities
```

3. Run the **db2ckmig** tool as follows for each database on your system:

```
db2ckmig database_name -L logfile -U username -P password
```

where:

- *database_name* is the name of your development database
- *logfile* is a path and filename for the log to be output to
- *username* is the user name for your development database

- *password* is the password for this user

In our example, we used the following:

```
db2ckmig mall -L D:\db2miglog.txt -U dbusr01 -P dbusr01pwd
```

If there are no potential problems, you will see the following output in the command line.

```
db2ckmig was successful. Database(s) can be migrated.
```

4. Inspect the content of the log file, specified as a parameter to **db2ckmig**, to ensure that no error conditions are logged.

If the tool reports any potential problems, you must resolve these before continuing with the remainder of this chapter.

5.5.3 Preparing a Windows user for DB2

In order to simplify the process of migrating the development database, it is advisable that you ensure that an administrative user, with the same user name and password as is used for DB2 on your VisualAge for Java V3.5.3, exists in Windows.

Also, if your WebSphere Commerce database was created under a different schema, you should create a user for that schema name.

Attention: While it is possible to use a different user name and password for the DB2 admin user and the WebSphere Commerce database owner, it is advisable that you use the same user for both in the development environment. This may help avoid configuration problems later on.

Create a Windows user with the appropriate access rights by completing the following steps:

1. Click **Start -> Settings -> Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Computer Management**.
4. In the tree on left of the new window, expand **Local Users and Groups** and click **Users**.
5. Select **Action -> New User...**
6. Enter the user name used for your original development database. In our example, we used dbusr01.
7. Enter the corresponding password in the Password and Confirm Password fields. In our example, we used dbusr01pwd.

8. Deselect **User must change password at next logon**.
9. Select **User cannot change password** and **Password never expires**.
10. Click **Create**.
11. Click **Close**.

Before this new user can be used by DB2, it must be given administrative rights. To do this, complete the following steps:

1. In the Computer Management window, right-click on the user you just created and select **Properties** from the menu.
2. Click the **Member Of** tab and click the **Add...** button.
3. Select **Administrators** from the list of user groups and click **Add**.
4. Click **OK** to add the user to the group and then click **OK** to close the user properties window.
5. Close the Computer Management window.

5.5.4 Installing DB2 Universal Database V8.1

Before installing DB2 Universal Database V8.1, you must ensure that the current installation is not in use and all DB2 services are stopped. To ensure this is the case, complete the following steps:

Note: At the time of writing, we had problems upgrading from DB2 Universal Database V7.2.5 to DB2 Universal Database V8.1.5 directly, using the CD supplied with WebSphere Commerce V5.6.

The following instructions documents how to install DB2 Universal Database V8.1.5 by installing the base product DB2 Universal Database V8.1 and then applying fixpack 5.

An alternative is to uninstall the DB2 Universal Database V7.2.5 installation and then install DB2 Universal Database V8.1.5.

1. Restart your computer.
2. Ensure that no local databases are in use. To check for connections, execute the following command in a DB2 command line window:

```
db2 list applications
```

The output from this command should be as follows:

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

If the response is a list of database agent connections, you must ensure that each of the corresponding processes are ended before continuing.

Tip: A typical installation of DB2 Universal Database V7.2.5 will include the DB2 Warehouse components. These will be registered as Windows services, to be started automatically. If the DB2 Warehouse services are running, the `list applications` command will return lines reporting that the database named DWCTRLDB is in use by the applications named IWH2LOG.EXE and IWH2SERV.EXE.

To eliminate these connections, stop the following services from the Services console, or by using the command line `net stop` command:

```
Warehouse logger
Warehouse server
```

3. Run the following command once you ensured that no databases are in:
`db2stop`
4. Ensure that you have no DB2 applications running, including DB2 Command Windows.
5. Open the Services console and stop all DB2 services, that is all services starting with the word DB2.

To install DB2 Universal Database V8.1, complete the following steps:

1. Run **setup.exe** from the root of the DB2 Universal Database V8.1 installation CD.
2. When the Launchpad windows opens, click **Install Products**.
3. Click **Next** to start the installation wizard.

Several warning messages will appear. In our example, we saw the following warnings:

- An earlier version of DB2 Universal Database is already installed as shown in Figure 5-1. We clicked **Yes** to continue.



Figure 5-1 Warning that an earlier version of DB2 Universal Database is installed

- Information that the Query Patroller Client is not available in the new version of DB2 Universal Database as shown in Figure 5-2. We clicked **Yes** to continue.

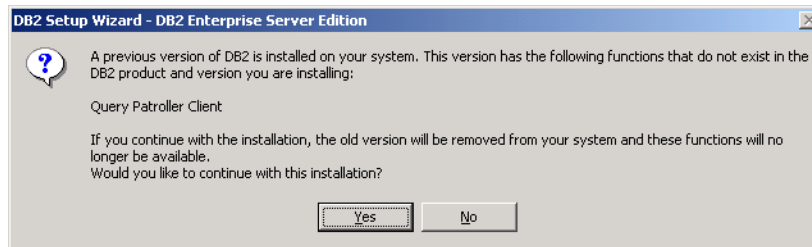


Figure 5-2 Warning that a feature is not available in DB2 Universal Database V8.1

- Information that we should run the db2chmig tool against our databases before upgrading, as shown in Figure 5-3. We clicked **OK** to continue, as we have already done this in 5.5.2, “Installation requirements” on page 78.

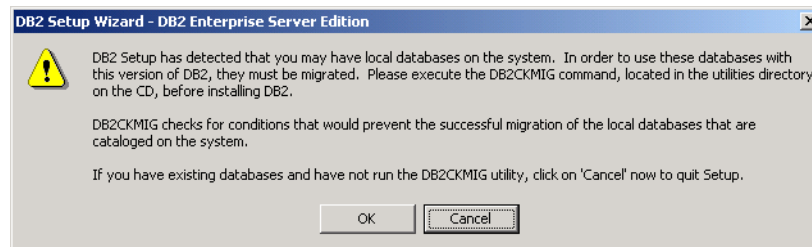


Figure 5-3 Warning that db2chmig should be run before installation

- On the welcome screen, click **Next**.
- Read the license agreement, select **I accept the terms in the license agreement** and click **Next**.
- Select the **Typical** option, uncheck **Data warehousing** and **Satellite administration capability**, and click **Next**.
- A warning about use of APPC will appear. This does not apply to us, so click click **OK** to continue.
- Ensure that **Install DB2 Enterprise Server Edition on this computer** is checked, that **Save your settings in a response file** is unchecked, and click **Next**.
- The path where DB2 Universal Database V7.2.5 is installed will be preselected. Ensure that you have sufficient disk space and click **Next** to continue.

10. In the User Information frame, enter the user name and password of the user that you created in 5.5.3, “Preparing a Windows user for DB2” on page 79. Ensure that **Use the same user name and password for the remaining DB2 services** is checked and click **Next**.
11. In the Administration contact list page, select **Local - Create a contact list on this system**, ensure that **Enable notification** is unchecked and click **Next**.

A warning window will appear to tell you that maintenance notifications will not be sent. Click **OK** to continue.
12. Unless you need to use the Task Center or the DB2 scheduler, ensure that **Do not prepare the DB2 tools catalog on this computer** is selected and click **Next**.
13. On the Specify a contact for health monitor notification page, select **Defer the task until after installation is complete** and click **Next**.
14. Review the settings in the text box and click **Install** to begin the installation.
15. When the window appears to tell you that the setup is complete, review the messages in the text area and click **Finish** when done.

If you are prompted to restart your computer, do so at this point.

After a few seconds, the **First Steps** window will appear. Close this window.

5.5.5 Installing DB2 Universal Database fixpack 5

The following steps detail how to install fixpack 5 for DB2. Before starting the installation, ensure that no databases are active and that all DB2 services are stopped. Refer to 5.5.4, “Installing DB2 Universal Database V8.1” on page 80 for details.

Tip: We encountered a problem with stopping the DB2 services on our machine. After issuing a `db2stop` command, we could not stop the DB2 - DB2CTLSV-0 service. As a workaround, we set all of the DB2 services startup types to Manual, noted the original settings and restarted the machine before installing the fixpack. Once installed, we reset the startup types to their original values.

1. Launch the fixpack installation executable, **FP5_WR21334_ESE.exe**.

The InstallShield Wizard installer window will start extracting the contents of the fixpack executable to a temporary location on your hard drive.

Important: You will need at least 500MB of free space on the drive where the TEMP environment variable points to. This is typically the Windows installation drive.

2. When the Launchpad window opens, click **Install Product**.

Note: The Launchpad window is identical to the original DB2 Universal Database V8.1 Launchpad window, although it is in fact the fixpack 5 Launchpad window.

3. Click **Next** to start the installation.

If a warning dialog is displayed warning you that DB2 is currently running, click **No** to exit the installation. Stop all running DB2 processes from the Services console and start the installation again.

Important: DB2 should be stopped cleanly before installing the fixpack. While it is possible to let the installation wizard kill the DB2 processes, this is not recommended as it could lead to unpredictable results.

4. On the welcome screen, click **Next** to continue.
5. After a while, the Wizard will indicate that setup is complete. Click **Finish** to exit the wizard.

After a few seconds, the **First Steps** window will appear.

6. Restart Windows.

You can confirm that the fixpack properly installed by opening a DB2 command window and running the **db2level** command. The output should be similar to that shown in Example 5-2.

Example 5-2 Sample output from db2level after applying fixpack 5

DB21-85I Instance "DB2" uses "32" bits and DB2 code release "SQL08015" with level identifier "02060106".

Informational tokens are "DB2 v8.1.5.449", "s040212", "WR21334", and FixPak "5".

Product is installed at "D:\WebSphere\sqllib".

5.5.6 Migrate databases to DB2 Universal Database V8.1 level

At this point, the databases in the system are all DB2 Universal Database V7.2.5 databases, not compatible with DB2 Universal Database V8.1. In order to use the databases, you must migrate these.

To migrate the databases, perform the following steps:

1. Log on to the system with a Windows administrator
2. In a DB2 Command Window, enter the following command for each of the databases in your system:

```
db2 migrate db <database_name>
```

Where <database_name> is the name of the database you wish to migrate.

When the migration is finished, You should see the following message:

```
DB20000I The MIGRATE DATABASE command completed successfully.
```

In our example, we entered the following commands:

```
db2 migrate db mall  
db2 migrate db pns  
db2 migrate db was  
db2 migrate db payman  
db2 migrate db satctldb  
db2 migrate db dwctrldb
```

Note: The **db2 migrate** command will only change the internal structures of the database to be compatible with DB2 Universal Database V8.1. No changes relating to the WebSphere Commerce schema in the database is changed. The WebSphere Commerce related migration is done in Chapter 7, “Migrating the development environment” on page 111.

5.6 Configuring the development environment for DB2

This section details the steps taken in order to configure the test environment to work with DB2 Universal Database. Additionally, the steps below will install the full test environment server. Once the steps in this section have been followed, you will have the option of using either the lightweight or the full test environment as both will be configured for DB2 Universal Database. For more information about the differences between the two, please refer to the *WebSphere Commerce Developer V5.6 Installation Guide for Windows*.

Note: After completing this chapter, the development database will still be using the schema for WebSphere Commerce Suite V5.1. The migration steps in the next chapter will migrate the database to the new schema for WebSphere Commerce V5.6.

Complete the following steps:

1. Ensure that WebSphere Studio Application Developer V5.1.1 is not running.
2. Open a DB2 Command Line Window by clicking **Start -> IBM DB2 -> Command Line Tools -> Command Window**.
3. Navigate to the following directory:

```
<wctoolkit_home>\bin
```

Where <wctoolkit_home> is the WebSphere Commerce V5.6 Toolkit installation directory. For example, in our example, we entered:

```
D:\
cd \WebSphere\WCToolkitPro56\bin
```

4. Run the following command to set the database type for the WebSphere Commerce test environment to DB2:

```
setdbtype db2 %db2path% database_name dbadminuser dbadminpass dbuser
dbpass
```

Where:

- *database_name* is the name of your original development database
- *dbadminuser* and *dbuser* are the user name created earlier
- *dbadminpass* and *dbpass* are the associated password for the above user

In our example, we used the following. Notice that we

```
setdbtype db2 %db2path% mall dbusr01 dbusr01pwd dbusr01 dbusr01pwd
```

The **setdbtype** script will add the full test environment to your WebSphere Commerce workspace in WebSphere Studio Application Developer V5.1.1 and configure both the full and the lightweight test environments to use DB2 as their database system.

At this point, the development environment is configured to use DB2 Universal Database, pointing to a WebSphere Commerce database with the WebSphere Commerce Suite V5.1 schema. The test server is thus not functional until you complete the migration steps in Chapter 7, “Migrating the development environment” on page 111.

5.7 Installing VisualAge for Java V4.0

If you have developed custom EJBs for your WebSphere Commerce Suite V5.1 store and wish to migrate them to your new development environment, you will need to install VisualAge for Java V4.0.

Attention: Please read the opening paragraphs of 7.5, “Migrating custom EJBs” on page 118 to determine if you wish to do this step.

If you are going to recreate your EJBs using the tools in WebSphere Studio Application Developer V5.1.1, which we recommend unless you have a large number of custom EJBs, you do not need to install VisualAge for Java V4.0 and can skip the remainder of this section.

The time requirements for the installation and configuration of VisualAge for Java V4.0 obviously depends on your hardware. The time we spend is summarized in Table 5-1.

Table 5-1 Time spent installing and configuring VisualAge for Java V4.0

Task	Time in minutes
Installing VisualAge for Java V4.0	30
Install features and fixes for VisualAge for Java V4.0	15
Prepare the VisualAge for Java V4.0 workspace	5
Import WebSphere Commerce Studio V5.1 repository	60

5.7.1 Installing VisualAge for Java V4.0

Follow these instructions to install VisualAge for Java V4.0. This process will take approximately 30 minutes to complete, depending on your hardware.

Note: VisualAge for Java V3.5.3 and VisualAge for Java V4.0 cannot coexist on the same system. As such, you must install VisualAge for Java V4.0 on a separate system, remote from your VisualAge for Java V3.5.3 installation.

Alternatively, if your access to additional systems is limited, you could perform the steps in the following order:

1. Follow the instructions in 6.2.4, “Export code from VisualAge for Java V3.5.3” on page 100 to export your custom Java code.
 2. Follow the instructions in 7.5.2, “Export the EJB project from VisualAge for Java V3.5.3” on page 120 to export your EJBs from VisualAge for Java V3.5.3.
 3. Uninstall VisualAge for Java V3.5.3.
 4. Install and configure VisualAge for Java V4.0 per the instructions in this section.
 5. Complete the migration steps in the remaining chapters of this part, skipping the export steps that you have already performed.
-
1. Insert the VisualAge for Java CD into your CD-ROM drive. If the VisualAge for Java launchpad does not open, it can be started by running the **setup.exe** file from the root of the installation CD.
 2. The VisualAge for Java launchpad will open. Click **Install Products**, and click **Install VisualAge for Java**.
 3. After a while the language selection window for the main installation program will appear. Select your language and click **OK**. We selected **English**.
 4. The InstallShield wizard starts up and the welcome window will appear. Click **Next**.
 5. Read the license agreement, select **I accept the terms in the license agreement** and click **Next**.
 6. The installation program will prompt for the installation type. Select **Complete** and click **Next**.
 7. The Edit Features window appears which will allow you to change the destination path folder. Click **Change...**
 8. Enter the desired destination paths for the components being installed and click **OK**.

We used the following value:

D:\WebSphere\VAJ40
 9. Click **Next**.

10. The repository location window appears. Ensure that **Local** is selected and click **Next**.
11. The Ready to install program confirmation window appears. Click **Install**.
12. When the installation is complete, a confirmation window will appear. Click **Finish** to close the window.

At this point, VisualAge for Java V4.0 is installed with a default workspace. In the next section, we will import the WebSphere Commerce Suite V5.1 repository, enabling the import and export of your customized EJBs.

5.7.2 Configuring VisualAge for Java V4.0

In order to import and compile your EJBs in VisualAge for Java V4.0, the WebSphere Commerce Studio V5.1 repository file must be imported into VisualAge for Java. Without this, you will not be able to export your EJBs as EJB 1.1 JAR files.

Before you can import the WebSphere Commerce Suite V5.1 repository into VisualAge for Java V4.0, you must apply fixes to VisualAge for Java V4.0 and prepare your workspace to make it compatible with the packages and classes that exists in the WebSphere Commerce Suite V5.1 repository.

Install features and fixes for VisualAge for Java V4.0

Before proceeding with the configuration, you must install the following fixes and features in the order listed:

- ▶ Install the DeployTool fix
- ▶ Add features to workspace
- ▶ Install the Readonly fix
- ▶ Install IBM WebSphere Test Environment fix
- ▶ Install IBM EJB Tool fix

The following sections will detail these steps.

Important: The following procedures performed from inside VisualAge for Java V4.0 must be executed using Administrator as the workspace owner unless otherwise specified.

To change workspace owner, do the following:

1. Click the **Workspace** menu, select **Change Workspace Owner**.
2. Select **Administrator** and click **OK**.

Install the DeployTool fix

The DeployTool fix upgrades the version of the Export Tool for EJBs and can be found on the original WebSphere Commerce Studio V5.4 CD in the `efixes/vaj` directory. The fix is stored in the file **EJB-1.1-DeployedTool.zip**.

To install this fix, complete the following steps:

1. Ensure that VisualAge for Java V4.0 is not running.
1. Extract the **EJB-1.1-DeployedTool.zip** file to a temporary directory on the VisualAge for Java V4.0 system.
2. Run **setup.exe** from the temporary directory and complete the instructions.

Add features to workspace

You must add various features to your VisualAge for Java V4.0 workspace before you can import the repository. To do this, complete the following steps:

1. Open VisualAge for Java V4.0. If this is the first time this program has been opened you will be prompted to assign the Windows user ID as the VisualAge for Java administrator. Enter the administrator user that you are currently logged in to Windows with.
2. If the **Welcome to VisualAge** window appears, uncheck **Show this window at startup**, select **Go to the Workbench** and click **OK**.
3. Press **F2** to get to the QuickStart panel.
4. From the left panel, select **Features**.
5. From the right panel, select **Add Feature**, and click **OK**.
6. Select the following features and click **OK**:
 - Export Tool for Enterprise Java Beans 1.1 4.1.5
 - IBM Common Connector Framework 3.5.3
 - IBM EJB Development Environment 3.5.3
 - IBM Java Record Library 3.5.3
7. Close VisualAge for Java V4.0.

Install the Readonly fix

Install the following fix to VisualAge for Java V4.0 to avoid VisualAge for Java V4.0 from deleting read-only attributes from your custom EJBs:

1. Ensure the VisualAge for Java V4.0 is not running.
2. Unzip the **readonly.zip** file from the `efixes\VAJ` of the WebSphere Commerce Studio V5.4 CD into your VisualAge for Java V4.0 directory.

This will create the following three files in the <vaj40_home>\ide\update directory, where <vaj40_home> is the installation directory for VisualAge for Java V4.0:

- readonly.tfs
 - readonly.tfi
 - readonly.txt
3. Open VisualAge for Java V4.0.
 4. Select **Workspace -> Tools -> Fix Manager**.
 5. In the Fix manager window, select **readonly - Prevent invalid deletion of EJB read-only attributes** and click the >> button, then click **OK**.
 6. Confirm that an * (asterisk) appears next to **readonly - Prevent invalid deletion of EJB read-only attributes**.
 7. Click **Close**.
 8. Close VisualAge for Java V4.0.

Important: You must restart VisualAge for Java in order for this fix to take effect.

Install IBM WebSphere Test Environment fix

From the **Projects** tab in VisualAge for Java V4.0, do the following:

1. Create an open edition of the IBM WebSphere Test Environment.
To create an open edition of a project, right-click on the project and select **Manage -> Create Open Edition**. If the **Create Open Edition** option is unavailable then the project is already in an open edition and you can ignore this step.
2. Right-click the **IBM WebSphere Test Environment** project and select **Import...**
3. Select **Jar file** and click **Next**.
4. Enter the path to the **PQ50159.jar** file in the filename field.
The **PQ50159.jar** file is placed in the `efixes\vaj` directory on the WebSphere Commerce Studio V5.4 installation CD.
5. Ensure that the **Project** field contains the text **IBM WebSphere Test Environment**.
6. Check **Create new/scratch editions of versioned projects/packages**.
7. Click **Finish**.

8. Version the IBM WebSphere Test Environment project by right-clicking on the **IBM WebSphere Test Environment** project, and selecting **Manage -> Version**.
9. Select **One Name**, enter 3.5.3 PQ50159 in the text field and click **OK**.

Install IBM EJB Tool fix

To apply the fix to the IBM EJB Tool project, do the following:

1. In VisualAge for Java V4.0, create an open edition of the project IBM EJB Tool.
2. Right-click the project again and select **Import...**
3. The Import Smartguide will appear. Select **Jar file** and click **Next**.
4. In the **Filename** field, enter the location of the **ivjfix35.jar** file. It is located in the following directory on the WebSphere Commerce Studio V5.4 CD:
 efixes\vaj
5. Ensure that **Create new/scratch editions of versioned projects/packages** is checked and click **Finish**.
6. Version the IBM EJB Tool project by right-clicking on the **IBM EJB Tool** project, and selecting **Manage -> Version**.
7. Select **One Name**, enter 3.5.3 ivjfix in the text field and click **OK**.

Prepare the VisualAge for Java V4.0 workspace

In order to be able to import the WebSphere Commerce Suite V5.1 repository into VisualAge for Java V4.0, you must delete the packages from the projects as shown in Table 5-2.

Table 5-2 Packages to delete from workspace

Project	Package	Version Name
Tcl	com.oroinc.regex	3.5.3 WCS 5.1 fix
IBM WebSphere Test Environment	javax.jms	3.5.3 WCS 5.1 fix

For each package in Table 5-2, create an open edition of the project it belongs to and then delete the package. When finished, version the project using the name suggested in the table.

Note: These instructions are only necessary if you choose to import the WebSphere Commerce Suite V5.1 repository into VisualAge for Java V4.0.

If you have access to the WebSphere Commerce Suite V5.4 repository and your custom EJBs do not rely on features only present in WebSphere Commerce Suite V5.1, you can use that repository instead, and will not have to delete the packages mentioned here.

To create an open edition of a project, right-click the project and select **Manage** -> **Create Open Edition**.

To delete a package select the package, right-click it and select **Delete...**, or press the **Delete** key, and click **Yes**.

Import WebSphere Commerce Studio V5.1 repository

You are now ready to import the WebSphere Commerce Studio V5.1 repository into VisualAge for Java. To do so, complete the following steps:

1. Extract the repository from the WebSphere Commerce Studio V5.1 CD to a temporary location on your hard drive.
2. Open VisualAge for Java V4.0.
3. From the **File** menu, select **Import**.
4. Select **Repository** and click **Next**.
5. Select **Local repository**.
6. In the **Repository name** text field, enter the location to which you extracted the WebSphere Commerce Suite V5.1 repository.
7. Select **Projects** and click **Details**. Check the following projects in the **Projects** list and click **OK**.
 - IBM WC Commerce Server
 - IBM WC Enterprise Beans
8. Ensure that the **Add most recent project edition to workspace** check box is selected and click **Finish** to begin importing.
9. Change the workspace owner to WCS Developer by doing the following:
 - a. From the **Workspace** menu, select **Change Workspace Owner**.
 - b. Select **WCS Developer** and click **OK**.
10. Save your workspace by selecting **Save Workspace** from the **File** menu.

A message will display when your workspace has been saved.
11. Click **OK** to continue.

VisualAge for Java V4.0 is now ready for you to import your custom EJBs as part of the migration. This will be covered in 7.5, “Migrating custom EJBs” on page 118.

6



Pre-migration steps

This chapter describes the tasks needed to prepare the development instance and database for migration. These steps will partially overlap the pre-migration steps for the runtime environment, covered in Chapter 10, “Pre-migration steps” on page 183.

This chapter is organized as following:

- ▶ Preparation overview
- ▶ Instance preparation
- ▶ Database preparation

6.1 Preparation overview

This chapter describes the actions needed to prepare the WebSphere Commerce Suite V5.1 development environment prior to migrating to WebSphere Commerce V5.6 Developer.

It is recommended to prepare and migrate the elements in the following order:

1. Instance
2. Database

The migration steps are described in Chapter 7, “Migrating the development environment” on page 111

The following is a list of the specific sections that are covered. You can use this list as a checklist when you are performing the tasks in this chapter. Read and understand the chosen section completely, before you begin any of the tasks:

- ▶ Instance preparation
 - Updating the product information file
 - Prepare resources
 - Merge EJB groups
 - Export code from VisualAge for Java V3.5.3
- ▶ Database preparation
 - Update database configuration values
 - Unsent messages
 - Order status
 - Catalog
 - Members
 - Custom message types
 - Dropping foreign key references
 - Erroneous data in encrypted fields

In each section we analyze the element to see what needs to be changed in order to prepare it for the migration.

6.2 Instance preparation

Before you can migrate the development instance, you must prepare the development system, as described in this section.

6.2.1 Updating the product information file

Before running the WCIM tool, you must create a **product.xml** file for WebSphere Commerce Suite V5.1. This file will be used by the tool to determine details about your installation that are required during the instance migration process.

Although WebSphere Commerce Suite V5.1.1.x provided a **product.xml** file, it is not in the structure required for migration. You must replace the file according to the instructions below. Versions prior to this did not have a **product.xml** file at all and thus one must be created.

To create the **product.xml** file, complete the following steps:

1. Copy the file **product.xml.51.sample** from the following directory:

`<wc_home>\migration`

To the following directory:

`<wcs_home>\xml`

where `<wc_home>` is the installation home directory for WebSphere Commerce V5.6 and `<wcs_home>` in the installation home directory for WebSphere Commerce Suite V5.1

2. Rename the file you just copied to **product.xml**.
3. Open the file **product.xml** in a text editor, for example, Notepad.
4. Update the relevant sections, shown in bold in Example 6-1 on page 98. Ensure that you enter the appropriate values for your scenario. Refer to Table 6-1, 6-2 and 6-3 on page 99 for assistance on how to choose the values for your environment.
5. Update the path node with the path to your WebSphere Commerce Suite V5.1 installation, such that it corresponds to the WebSphere Commerce Suite V5.1 installation path specified in the `WCInstallDir` attributes of the Instance node in the *instance.xml*.

Note: No other tags should be updated as they will be updated by the WCIM tool during instance migration. Ignore the comments in the sample file, telling you to update several of the other values in the file.

- 6. Save and close the file.
- 7. Copy the file **product.dtd** from the following directory :
 <wctoolkit_home>\xml
To the following directory:
 <wcs_home>\xml
For example, we copied the file **product.dtd** from the directory:
 D:\WebSphere\WCToolkit56\xml
To the following directory:
 D:\WebSphere\WCS\xml

Example 6-1 Example product.xml file

```
<websphere>
:
<commercesuite>
:
  <commerceserver>
    <name>IBM WebSphere Commerce Server</name>
    <edition>
      <name>edition_name</name>
    </edition>
    <version>5</version>
    <release>1</release>
    <modification>mod</modification>
    <fixpak>fixpak</fixpak>
    :
    <install>
      :
      <path>path</path>
    :
    </install>
    :
  </commerceserver>
  :
</commercesuite>
</websphere>
```

Table 6-1 Choosing the value for edition_name

WebSphere Commerce version	edition_name value
WebSphere Commerce Suite V5.1 Start Edition	Start

WebSphere Commerce version	edition_name value
WebSphere Commerce Suite V5.1 Professional Edition	Pro
WebSphere Commerce Suite V5.1 Business Edition	Business

Table 6-2 Choosing the value for mod

WebSphere Commerce version	mod value
WebSphere Commerce Suite V5.1.0.1	0
WebSphere Commerce Suite V5.1.1	1
WebSphere Commerce Suite V5.1.1.x	1

Table 6-3 Choosing the value for fixpack

WebSphere Commerce version	fixpack value
WebSphere Commerce Suite V5.1.0.1	1
WebSphere Commerce Suite V5.1.1.1	1
WebSphere Commerce Suite V5.1.1.2	2
WebSphere Commerce Suite V5.1.1.3	3
WebSphere Commerce Suite V5.1.1.4	4

6.2.2 Prepare resources

The WebSphere Commerce V5.6 migration script will import properties files, containing language independent JSP texts, into WebSphere Studio Application Developer V5.1.1 from the following directory and subdirectories thereof:

```
<wcs_home>\stores\properties\<instancename>
```

Where *instancename* is the name of your WebSphere Commerce instance and <wcs_home> is the base WebSphere Commerce Suite V5.1 installation directory.

JSPs will be extracted from the following directory and subdirectories thereof:

```
<vaj_home>\ide\project_resource\IBM WebSphere Test Environment\hosts\default_host\default_app\web
```

Where <vaj_home> is the VisualAge for Java V3.5.3 installation directory.

In order for the migration tools to correctly migrate properties files and JSPs, you must ensure that they are located in the directories mentioned.

Note: It may be that the properties and JSP files for your store already exist in the mentioned directories. In either case, you should double-check that these files are the correct versions.

6.2.3 Merge EJB groups

The WebSphere Commerce V5.6 Toolkit expects all custom EJBs to be defined in a single EJB project, named WebSphereCommerceServerExtensionsData. Since WebSphere Studio Application Developer does not handle the importing of several EJBs from different EJB 1.1 JAR files, it is important that all your custom EJBs can be exported into a single EJB 1.1 JAR file.

In 7.5.5, “Export EJBs as EJB 1.1 JAR files” on page 122 the EJBs will be exported from VisualAge for Java V4.0, which only supports exporting each EJB group into their own EJB 1.1 JAR file. This means that before exporting your custom EJBs, you must ensure that all custom EJBs are defined in a single EJB group in VisualAge for Java.

This also implies that all EJB code is isolated into one project in VisualAge for Java V3.5.3. It is important that this project only contain EJB related code. That is, EJB project does not contain any WebSphere Commerce command or data bean code.

Ensure that you version the project containing your EJBs before continuing. Failure to do so will mean that you will not be able to export the project in the next step. To version your project follow these steps:

1. Start Visual Age for Java 3.5
2. In the project list, select the project and right click on it.
3. Select **Manage -> Version**, if the command is not available, it has been versioned. If the command is available, version the project by clicking on Version.

6.2.4 Export code from VisualAge for Java V3.5.3

In this step, any custom code or extensions to existing WebSphere Commerce commands and data beans must be exported from VisualAge for Java. We are not concerned with EJBs at this stage, only command code, data beans and utility classes etc. The procedure for migrating EJBs is described in 7.5,

“Migrating custom EJBs” on page 118 and 8.2.6, “Configuration and code changes for migrated EJBs” on page 144.

Overview

VisualAge for Java provided a source control feature through the concept of a *repository*. This feature allowed typical source control functionality such as restoring previous editions of code and marking releases etc.

VisualAge for Java also has the concept of *projects* that are used to logically group code for applications or components. These projects, and the packages and files within them, could be versioned in the repository.

WebSphere Studio Application Developer V5.1.1 does not include any source code management features as standard. However, various plug-ins are widely available to allow the use of external systems, such as CVS.

As the migration script cannot cater for all possible source control solutions under WebSphere Studio Application Developer V5.1.1, it does not migrate all historic versions of code from the VisualAge repository. Only the code which is currently available in the workbench will be ported.

Tip: If you need to migrate all versions of your code to your new environment, you can do this by manually exporting the source code for each edition from VisualAge for Java, importing this into WebSphere Studio Application Developer, and then commit the imported source code to the new source control management system.

As this is a very tedious and lengthy process, we recommend that you keep at least one installation of VisualAge available to access previous versions of code and only export the latest version to WebSphere Studio Application Developer V5.1.1.

Exporting the Java code

The WebSphere Commerce V5.6 migration script requires that all of the custom code source that is to be migrated must be packed in a single JAR file. You will be prompted to enter the location of this JAR when the script is run.

In order to export your code, locate the projects in VisualAge where it resides and complete the following steps:

1. With the **Projects** tab selected, select all of the projects that contain your custom code.
2. Right-click on one of these selected projects and click **Export...** from the menu.

3. Select **Jar file** and click **Next**.
4. In the **Jar file:** text box, enter a path and filename for your exported code. In our example, we used `D:\MyStoreCommands.jar`.
5. Ensure that only the **.java** check box is checked, and unchecking the **.class**, **resources** and **beans** if necessary.

If you wish to check that all of your code is selected correctly, you can click the **Details...** button next to **.java**.
6. Ensure that **Do not seal the jar** is selected and click **Finish**.

Tip: In VisualAge, you were required to import utility or library classes into the workspace from the JAR files they were packed in.

In WebSphere Studio Application Developer V5.1.1, you can reference the original library JAR files directly within the project properties.

6.3 Database preparation

This section describes the tasks needed to prepare the database for the WebSphere Commerce V5.6 database migration script. Each task is split up in subsystems and can be applied in any order.

It is assumed that you have already migrated the internal DB2 structures to make them compatible with DB2 Universal Database V8.1, as described in 5.5.6, “Migrate databases to DB2 Universal Database V8.1 level” on page 85.

As a test database was not available to us at the time of writing, we opted to use a copy of the production database. In order to keep the size of the database as small as possible, we deleted all guest users and pending orders from the database before starting the migration. This was intended to shorten the time that the various tools and scripts would take to run.

6.3.1 Update database configuration values

The database configuration values, such as logfile size, application heap and statement heap, must be updated up to WebSphere Commerce V5.6 level prior to running any migration scripts. A script file is provided with WebSphere Commerce V5.6 for updating the database configuration values:

```
<WCToolkit_home>\bin\updateDB2Configuration.bat
```

The script is invoked from a DB2 Command Window within the `<wc_home>\bin` directory as follows:

```
updateDB2Configuration <database_name> [<logfile>]
```

Where <database_name> is the name of the database to change configuration values for and <logfile> is an optional parameter, naming the file to use for logging the progress of the script.

Important: The script must be called from a DB2 Command Window. If the script is run from a regular Windows command prompt window, the following output will be shown 11 times in the console, or log file if specified:

```
DB21061E Command line environment not initialized.
```

In our example we called the as follows:

```
D:
cd \WebSphere\WCToolkitPro56\bin
updateDB2Configuration mall c:\logs\updatedb2configuration.log
```

Once the script has executed, check the logs to ensure no errors has occurred. The script runs a series of SQL commands, so any SQL command failing is an indication of an error.

The script will update the database configuration values with the minimum recommended as shown in Table 6-4. Base on your database characteristics, your DBA should review and modify these values accordingly in the script before executing the script.

Table 6-4 DB2 configuration set by the updateDBConfiguration script

Parameter	Value
applheapsz	1000
stmtheap	60000
app_ctl_heap_sz	8192
locklist	2400
indexrec	RESTART
logfilsize	1000
logprimary	12
logsecond	10
pckcachesz	4096
catalogcache_sz	4096

Important: Ensure that all connections to the database are terminated before continuing. Otherwise the changes performed by the script in this section may not take effect.

6.3.2 Unsent messages

If the development database contains any messages in the MSGSTORE table, these must be removed before migrating the database, as they cannot be handled by WebSphere Commerce V5.6.

As this is a development database, there is no reason to attempt to have these messages sent and they can safely be deleted.

The following SQL will clean the MSGSTORE table:

```
delete from MSGSTORE
```

6.3.3 Order status

Any orders in state C (Payment approved) will result in an extra allocation of the products in those orders from the inventory when the migrated instance is restarted.

As this is a development database, this may not be an issue. If it is a potential problem, we recommend that you do one of the following:

- ▶ Delete the orders:

```
delete from order where status='C'  
delete from ordeitems where status='C'
```

- ▶ Change the order status to S (Shipped):

```
update order set status='S' where status='C'  
update ordeitems set status='S' where status='C'
```

- ▶ Change the order status to X (Cancelled):

```
update order set status='X' where status='C'  
update ordeitems set status='X' where status='C'
```

6.3.4 Catalog

This sections describes the changes which needs to be applied for the catalog subsystem.

Master catalog

In order to be able to use the product management tools in the WebSphere Commerce Accelerator, the store must have a valid master catalog.

Refer to “Master catalog” on page 193 for details about master catalogs and a procedure for generating a valid master catalog for your store.

Product-item relationship

In WebSphere Commerce V5.6 all items must have a product. This rule is enforced as the Product Management Tool uses products as its starting point.

This rule can be applied in two ways (if needed):

1. Manually create products and associate items.

The benefits with this approach is less products and more organized product-item relations.

2. Let the migration script create one product (placeholder) for each item.

The benefits of this approach is that the script deals with creating the products, but the user may need to reorganize the products (placeholders) and their items after the migration.

In our example we let the migration script create the products (placeholders) for the items.

6.3.5 Members

This section describes the changes which needs to be applied for the member subsystem. This includes:

- ▶ Organizational structure
- ▶ Profile types
- ▶ Administrators
- ▶ Role changes

Organizational structure

Every user and organizational entity in WebSphere Commerce V5.6, with the exception of Root Organization, must have a parent organizational entity. This allows users and organizational entities to form a membership hierarchy.

The migration script will ensure that this is enforced, using the following rules:

- ▶ All organizations without a parent organization, that is organizations for which the MEMBER_ID column of ORGENTITY is set to null, will have Root Organization assigned as their parent organization.

If this is not desired, you must update the MEMBER_ID column for the organizations that must have other parent organizations before migrating.

- ▶ All users without an entry in the BUSPROF table will have Default Organization assigned as parent organization.

If this is not desired, you must update create a business profile, containing references to the desired parent organization, for the relevant users before migrating.

Also, remember to set the profile type to B (Business user) for the relevant users.

- ▶ Users with an entry in BUSPROF and with PROFILETYPE set to B (Business user) will be assigned the organization, pointed to by the ORGUNIT_ID or ORG_ID columns, as follows:
 - If ORGUNIT_ID is not null, the organization pointed to by ORGUNIT_ID, is used. as the user's parent organization
 - If ORGUNIT_ID is null and ORG_ID is not null, the organization pointed to by ORG_ID, is used as the user's parent organization.
 - If both ORGUNIT_ID and ORG_ID are null, the user is assigned Default Organization as parent organization.

Since business users should not be assigned Default Organization, we recommend that you ensure that all your business users an entry in BUSPROF with the right organization specified in ORGUNIT_ID.

Profile types

The WebSphere Commerce Suite V5.1 and WebSphere Commerce V5.6 schema allows for three profile types, as defined by the PROFILETYPE column of the USERS table:

- ▶ C - Consumer (B2C)
- ▶ B - Business (B2B)
- ▶ null - No profile data

It is recommended to clean up this column such that all users have either a Consumer or a Business profile, that is to say: ensure that no rows in the USERS table have the PROFILETYPE column set to null.

Administrators

Administrators with register type S or A (Site administrator or Administrators, respectively) that also belongs to an access group must have their profile type set to B (Business user).

This is because all future administrators created in Organization Administration Console will have profile type B. This can be done with the following SQL:

```
update users set profiletype='B'
where registertype in ('A','S')
and users_id in (select users_id from accmbrgrp)
```

Role changes

Some of the roles in WebSphere Commerce Suite V5.1 are no longer available in WebSphere Commerce V5.6. Table 6-5 lists the WebSphere Commerce Suite V5.1 and the roles that should be used in WebSphere Commerce V5.6.

Table 6-5 Roles that are not included in WebSphere Commerce V5.6

WebSphere Commerce Suite V5.1 role	WebSphere Commerce V5.6 role
Order clerk (-5)	Customer Service Supervisor (-14)
Store Developer (-7)	Site Administrator (-1)
Customer role	Refer to WebSphere Commerce V5.6 migration guide for more information
Merchant	Seller
Merchandising Manager	Product Manager

Refer to *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1* for more information on the details regarding role migration.

The recommended approach to the roles, that are not included in WebSphere Commerce V5.6, is to replace them with roles which have the same function.

If a role is found by the migration script, and not found in WebSphere Commerce V5.6, it is migrated as a user-defined role.

Our application had two users with the following roles which needed to be changed:

1. Order Clerk
2. Merchandising Manager

3. Customer Service Representative

We found that Store Administrator would be a more suitable role for these two users. Therefore we assigned these two users the Store Administrator role instead.

Delete all roles for user:

```
delete from accmbrgrp where users_id=<users_id>
```

Add Store Administrator as role for our users:

```
insert into accmbrgrp values (-6,<users_id>,-2000,NULL)
```

Where <users_id> is the ID of the users that should be assigned the Store Administrator role.

6.3.6 Custom message types

If you have created any custom message types in the MSGTYPES table, you may be required to update the MSGTYPE_ID column for your new types in order to avoid a conflict with new message types introduced in WebSphere Commerce V5.6. If you have not created any custom types, you can safely skip this sub-section.

As the MSGTYPE_ID column in the MSGTYPES must contain unique values, you must ensure that any custom message types that you have created do not have a MSGTYPE_ID the same as any new types in WebSphere Commerce V5.6. The simplest way to ensure that this does not happen is to update your custom types by adding 10000 to MSGTYPE_IDs. This must be done manually from a DB2 Command Window or the DB2 Command Center.

Failure to complete this step for custom message types can result in the database migration script failing to complete successfully.

6.3.7 Dropping foreign key references

If you have created any custom tables that have foreign key references to tables in the standard WebSphere Commerce data model, you must drop these foreign keys before proceeding with the migration. During the database migration, many tables are dropped, replaced with new versions and then re-populated with your data. If foreign key relationships exist, this may prevent the tables from being dropped, leading to the database migration script failing.

You must also drop any custom views that have been created in the database, as these may interfere with the operation of the database migration scripts.

Once the migration has completed successfully, you must recreate your foreign key relationships manually. If possible, while you are working on the migration, it is a good idea if a member of your team starts create a script that will recreate the references once the migration has completed. They will need to refer to the online help to check that the data model changes do not impact the referential integrity or appropriateness of the custom tables. The online help can be found at:

<http://publib.boulder.ibm.com/infocenter/wc56help/index.jsp>

6.3.8 Erroneous data in encrypted fields

Before migrating encrypted data, you should check the following fields in Table 6-6 for data that has a leading space and is unencrypted.

In WebSphere Commerce, in a field that may be encrypted, such as the field listed in Table 6-6, the presence of a leading space indicates that the field contains encrypted data.

If any of your unencrypted data has a leading space, due to a user entering this in the store for example, the migration tool will misinterpret this and try to decrypt the value, causing an error. The tables and fields to check are shown in Table 6-6. For further information about the behavior of the encrypted data migration script, please refer to Appendix C, "Migration scripts" on page 255.

Table 6-6 Tables and fields to check for erroneous leading spaces

Table	Columns to check	Notes
ORDPAYINFO	VALUE	Credit Card related information
ORDPAYMTHD	PAYDEVICE	Credit card related information



Migrating the development environment

This chapter describes the process of migrating your store from the WebSphere Commerce Suite V5.1 development environment with VisualAge for Java V3.5.3 to the WebSphere Studio Application Developer V5.1.1 based WebSphere Commerce V5.6 development environment.

The chapter is organized in the following sections:

- ▶ Current development environment
- ▶ Development environment migrated
- ▶ Migrating the development instance
- ▶ Migrating the development database
- ▶ Migrating custom EJBs

7.1 Current development environment

This is a high level overview of the development environment as was installed for WebSphere Commerce Suite V5.1 development.

Our WebSphere Commerce Suite V5.1 development environment consisted of a single-tier installation of WebSphere Commerce running on DB2 Universal Database V7.2.5. We had a single instance called *demo* created for our store. Payment Manager was not installed as the store utilized an external payment provider's API. A library JAR file provided by Verisign was extracted and imported into the VisualAge workspace.

The development tools consisted on VisualAge for Java and WebSphere Commerce Studio. The WebSphere Test Environment was configured in VisualAge.

7.2 Development environment migrated

Figure 7-1 shows an overview of the development environment as it will be after the migration. As the figure shows, both environments will be functional after the migration is completed. The EJB Migration Node, shown in Figure 7-1 in a dashed box, is an optional node used only if you choose to migrate your custom EJBs using the process described in this book. As noted in 7.5, "Migrating custom EJBs" on page 118, the migration process is not straightforward and is not recommended unless you have a large number of custom EJBs.

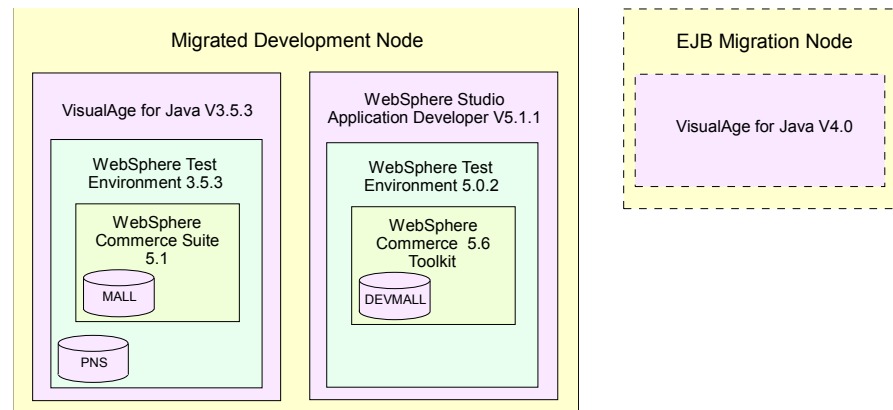


Figure 7-1 Schematic overview of the migrated development environment

7.3 Migrating the development instance

This section details the use of the WCIM tool to perform the instance migration in your development environment. In WebSphere Commerce Suite V5.1, the development environment is reliant on a local run-time instance. Configuration of the WebSphere Test Environment in VisualAge is taken from the configuration files of this instance. During migration of the development environment to WebSphere Studio Application Developer V5.1.1, this configuration must be read and duplicated in the new environment.

7.3.1 Overview

The WCIM tool performs this instance migration in a very similar manner to that of a run-time migration. The following is an overview of its behavior:

- ▶ The *instance.xml* file is migrated.
- ▶ Store front assets are copied from VisualAge into a Web project in WebSphere Studio Application Developer V5.1.1. This includes any JSPs, HTML, CSS files, images etc. that are within the WebSphere Test Environment. The files are copied from the following directory and its subdirectories within the VisualAge installation directory:

```
<vaj_home>\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\web
```

JSPs are updated to conform to the JSP 1.2 specification as is required by the new version of WebSphere Commerce.

- ▶ Properties files used for national language support are copied from the WebSphere Commerce run-time to the new development environment. The files are copied from the following directory and subdirectories thereof:

```
<wcs_home>\stores\properties
```

- ▶ Custom code is extracted from the JAR file exported from VisualAge and is imported into the WebSphereCommerceServerExtensionsLogic project in WebSphere Studio Application Developer V5.1.1.

Important: Before running the instance migration tool, you must ensure that neither VisualAge for Java nor WebSphere Studio Application Developer V5.1.1 are running.

Also, we recommend that you perform a full system backup of the WebSphere Commerce Suite V5.1 development system before proceeding.

7.3.2 Running the tool

To run the WCIM tool, complete the following steps:

1. Open a command prompt in windows.
2. Navigate to the following directory:

```
<wctoolkit_home>\bin
```

In our example, we used the following directory:

```
D:\WebSphere\WCToolkitPro56\bin.
```

3. Run the WCIM tool as follows:

```
wcim -component toolkit -from 51 -instance instancename
```

where *instancename* is the name of your instance in your development environment. In our example, we ran the command as follows:

```
wcim -component toolkit -from 51 -instance demo
```

4. You will be prompted to enter the path to the home directory of WebSphere Commerce. You must enter the path to the WebSphere Commerce Suite V5.1 installation on your development machine.

For example, we entered D:\WebSphere\WCS.

5. You will be prompted to enter the path to the home directory VisualAge for Java V3.5.3. Enter the full path to the file.

For example, we entered D:\WebSphere\VAJava.

Next, you will be prompted to provide the name of the custom code JAR file. You must enter the full path and file name of the JAR file you exported in 6.2.4, “Export code from VisualAge for Java V3.5.3” on page 100.

For example, we entered D:\MyStoreCommands.jar.

The tool will now complete the remaining steps with no further input. Example output from the tool in the Command Window can be seen in Example 7-1.

Ensure that the output from the tool reports no errors. On our system, the tool took approximately one minute to finish.

Example 7-1 Sample output from the WCIM tool

```
Event: MethodId: parseArgs
Event: MethodId: initLogFile
Event: *****
Event: * WCIM was started at 2004.06.09.22.45.51.000075 *
Event: *****
Event: MethodId: getWCIMPlan - getting WCIM migration plan
Event: MethodId: getAllCommands
Event: MethodId: shortenCmdList
```

```

Event: Starting to execute command
com.ibm.commerce.migration.wcim.command.WCIMMigrateStudioCommand
Event: Method: migrateDevEnv
Enter the previous WebSphere Commerce home directory: D:\WebSphere\WCS
Enter the home directory of VisualAge for Java: D:\WebSphere\VAJava
Enter the name of the custom code JAR file: D:\MyStoreCommands.jar
Event: MethodId: updateProductXML
Event: MethodId: updateProductXML
Event: MethodId: setNewVersionsEditions
Event: MethodId: invokeAnt - executing ant task: MigrateStudioInstanceBoth
Event: MethodId: setOldVersionsEditions
Event: MethodId: migrateConfigs - migrating demo.xml
Event: MethodId: getInstMigrateXmIs - getting XML files to update instance.xml
Event: MethodId: getInstanceUpdatePath
Event: WCIM has successfully completed command
com.ibm.commerce.migration.wcim.command.WCIMMigrateStudioCommand
Event: WCIM has completed the job(s) assigned successfully.

```

When the WCIM tool has completed, you should check the WCIM logs located in the following directory:

```
<wctoolkit_home>\temp\logs
```

The file name of the log will be in the following format:

```
wcim.migration.toolkit.instanceName.51.YYYY.MM.DD.HH.MM.SS.ssssss.log
```

Where instanceName is the name of the instance and YYYY.MM.DD.HH.MM.SS.ssssss is a timestamp.

When you have validated that the migration was successful, you should open WebSphere Studio Application Developer V5.1.1 and refresh your workspace in order to make the new imported code visible.

To start WebSphere Studio Application Developer V5.1.1 with the WebSphere Commerce V5.6 Toolkit, select **Start -> Programs -> IBM WebSphere Commerce Developer Professional Edition -> WebSphere Commerce development environment**.

To refresh the WebSphereCommerceServerExtensionsLogic project, right-click the project and select **Refresh**.

Note: At this point the WebSphere Test Environment inside WebSphere Studio Application Developer V5.1.1 has been updated and your custom commands and data beans have been imported into WebSphere Studio Application Developer.

The development environment is not completely migrated until the migration steps in the following section are completed.

7.4 Migrating the development database

This section details how to migrate your development database. It is assumed that you have already upgraded DB2 Universal Database V7.2.5 to DB2 Universal Database V8.1 on your development system as detailed in 5.5, “Installing DB2 Universal Database V8.1” on page 77. This section explains how to update your existing database so that it can be used in DB2 Universal Database V8.1 and how to migrate it to the new schema for WebSphere Commerce V5.6.

7.4.1 Migrating unencrypted data

This section details the steps that should be taken to migrate your existing development database from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

To perform the migration, complete the following steps:

1. Open a DB2 Command Window if one is not already open.
2. Navigate to the following directory:

```
<wctoolkit_home>\bin
```

3. Run the migration script as follows:

```
migratedb -dbtype db2 -dbname database_name -dbuser user -dbpass pass  
-from 51 -instance instancename -schema schema
```

where

- *database_name* is the name of your development database
- *user* is the user name for your database
- *pass* is the associated password
- *instancename* is the name of your instance
- *schema* is the schema name for your WebSphere Commerce tables, which is usually the same as the database user name

We used the following command:

```
migratedb -dbtype db2 -dbname mall -dbuser dbusr01 -dbpass dbusr01pwd  
-from 51 -instance demo -schema dbuser01
```

In our example, with 1400 users, 1400 orders and 2700 order items, the database migration script took 15 minutes.

7.4.2 Migrating encrypted data

This sections explains how to run the script to migrate your encrypted data within your database and to reflect a new merchant key. This allows you to log on to the tools and your store using existing users and passwords.

To run the script to update your encrypted data, complete the following steps:

1. Ensure that WebSphere Studio Application Developer V5.1.1 is not running.
2. Open a DB2 Command Window.
3. Navigate to the following directory:

```
<wctoolkit_home>\bin
```

For example, we used:

```
D:\WebSphere\WCToolkitPro56\bin
```

4. Run the script as follows:

```
MigrateEncryptedInfo db2 instancename currentkey newkey
```

Where:

- *instancename* is the name of your instance
- *currentkey* is the key that was used in your previous development environment database. This is optional.
- *newkey* is the new merchant key you used in your new development environment. This is optional.

Note: It is most likely that you will not need to specify a new merchant key as the merchant key from your WebSphere Commerce Suite V5.1 development environment will have been migrated to WebSphere Studio Application Developer V5.1.1. In our example, we used the database from the production system which had a different merchant key to that which we specified during our WebSphere Commerce Suite V5.1 installation used for development. Because of this, we needed to specify this new key.

It is important that you still complete this step, omitting the *currentkey* and *newkey* arguments when running the script.

7.4.3 Summary

At this point, you should be at the following stage of migration:

- ▶ You have installed and configured the new WebSphere Commerce test environment in WebSphere Studio Application Developer V5.1.1.
- ▶ Your development WebSphere Commerce instance is migrated to the new test environment in WebSphere Studio Application Developer V5.1.1.
- ▶ Your existing development database has been migrated from DB2 Universal Database V7.2.5 to DB2 Universal Database V8.1.
- ▶ Your development database has been migrated to the new schema for WebSphere Commerce V5.6.
- ▶ Any encrypted data will be re-encrypted using the new algorithm and new merchant key (if required).

Before your store will work correctly in the new development environment, there are several steps that must be taken to modify your custom code. These steps are explained in the following chapter. If you have any custom EJBs you will now need to migrate these to WebSphere Studio Application Developer V5.1.1.

Important: If you dropped any foreign key constraints before performing the migration, you must remember to recreate these before testing your migrated store.

7.5 Migrating custom EJBs

The EJBs in WebSphere Commerce Suite V5.1 complied to the EJB 1.0 standard. In WebSphere Commerce V5.6 EJBs should conform to the EJB 1.1 standard. While WebSphere Application Server V5.0.2 is capable of running EJBs at either level, it is not recommended that EJBs of different specifications are mixed. As a result, any existing EJBs that you may have must be migrated to the new specification.

Note: The process of migrating EJBs to the new specification is not straightforward. The level of effort involved is often the same or greater than recreating the EJBs from scratch using the tools in WebSphere Studio Application Developer V5.1.1.

We would recommend that unless you have 10 EJBs or more you should consider recreating them in WebSphere Studio Application Developer V5.1.1. If you have session beans, you can simply copy the business logic code into your newly created EJBs.

For information about creating EJBs in WebSphere Studio Application Developer V5.1.1, refer to *SG24-6957: WebSphere Studio Application Developer Version 5 Programming Guide*.

In order to continue with the EJB migration, you must have installed VisualAge for Java V4.0 as detailed in 5.7, “Installing VisualAge for Java V4.0” on page 87.

7.5.1 Overview

The migration of the custom EJBs is a two stage process. VisualAge for Java V3.5.3 does not have the ability to convert EJBs to the EJB 1.1 specification. Unfortunately, WebSphere Studio Application Developer V5.1.1 cannot correctly import EJB 1.0 code and migrate it to the EJB 1.1 specification either. To work around this issue, the code must be migrated via an intermediate step. The code must be imported into VisualAge for Java V4.0 which does have the capability to export EJB 1.0 code in to the EJB 1.1 specification.

The process is outlined in Figure 7-2. The following sections will describe the process of exporting and importing EJBs:

- ▶ Export the EJB project from VisualAge for Java V3.5.3
- ▶ Regenerate the deployed code for the EJBs in VisualAge for Java V4.0
- ▶ Export EJBs as EJB 1.1 JAR files
- ▶ Import the EJBs into WebSphere Studio Application Developer

After the EJBs have been imported into WebSphere Studio Application Developer V5.1.1, there are still a number of code changes that must be performed before the migration is complete. These code changes are described in 8.2.6, “Configuration and code changes for migrated EJBs” on page 144.

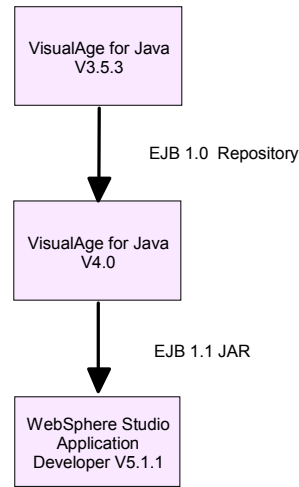


Figure 7-2 Process for converting from EJBs from version 1.0 to 1.1

Before you continue, you must have installed and configured a VisualAge for Java V4.0 system as described in 5.7.1, “Installing VisualAge for Java V4.0” on page 87.

Note: There may be other tools available to migrate EJBs between the two specification levels that you may consider. However, it is unlikely that these tools would successfully be able to compile the code and verify it as there are dependencies on WebSphere Commerce libraries.

7.5.2 Export the EJB project from VisualAge for Java V3.5.3

Once you have installed and configured the VisualAge for Java V4.0 remote node you may continue migrating your custom EJBs, the following step is to export EJBs from VisualAge for Java V3.5.3. There are several ways you could export the code, but the simplest method for importing into VisualAge for Java V4.0 is to export as a VisualAge repository.

In order to export the EJB code, complete the following steps:

1. If not already open, launch VisualAge for Java V3.5.3.
2. Select the **Projects** tab in the Workbench
3. Select the project that contains custom EJB code from the All Projects pane.

Reminder: Ensure that only EJB code exists in the project you select for export.

4. Right-click on the project and select **Export...** from the menu.
The Export Smart Guide will appear
5. Select **Repository** as the export destination and click **Next**.
6. On the next page, make sure that **Local Repository** is selected.
7. Specify a location for your exported repository in the **Repository name** text-box. For example, D:\myStoreEjbs.dat
8. Ensure that **Projects** is selected from the list of export options. You can confirm that the correct project and its versions are selected by clicking the **Details...** button.
9. When you are sure that the correct project, containing your EJB code, is selected, click **Finish**.
10. If you are prompted to create the new repository file, click **Yes**.
Your code will now be exported. This can take several minutes depending on how many EJBs you have in your projects.

Once the code has been exported, you can close VisualAge for Java V3.5.3. You should now either copy the exported repository file to the system where you installed VisualAge for Java V4.0 or ensure that the file is visible from the system.

7.5.3 Import the EJB projects into VisualAge for Java V4.0

The next step is to import the repository containing your EJB code into VisualAge for Java V4.0.

To import the repository, complete the following steps:

1. If not already open, launch VisualAge for Java V4.0.
2. Click the **File** menu and select **Import...**
3. In the Import SmartGuide, select **Repository** as the import source and click **Next**.
4. In the Import from another repository page:
 - a. Select **Local repository**.
 - b. Enter the path and filename of the repository file you exported in the last section in the **Repository name** text box.

- c. Select the option to import **Projects** and click **Details...** next to this option.
- d. In the window that opens, select your EJB project and click **OK**.
- e. Check the option to **Add most recent project edition to workspace** and click **Finish**.

Even with only small numbers of EJBs in your repository, VisualAge can take several minutes to complete the import.

5. Check that your project and its EJBs have appeared in the Projects and EJB views of the Workbench.

Your EJBs are now imported into VisualAge for Java V4.0, but they still only comply to the EJB1.0 specification. Additionally, the deployed code will need to be regenerated as will be shown in the following section.

7.5.4 Regenerate the deployed code for the EJBs

Before exporting your EJBs in the EJB 1.1 format, you must regenerate the deploy code. This can be done for all of your custom EJBs in one step.

To regenerate the deploy code, complete the following steps:

1. If it is not already open, launch VisualAge for Java V4.0.
2. Switch to the EJB view in the Workbench by clicking the **EJB** tab.
3. Select the EJB group that contain your custom EJBs in the Enterprise Beans pane.
4. Right-click on one of the selected groups and select **Generate Deployed Code** from the menu.

A dialog will appear showing you the progress. If you have a lot of beans, this may take several minutes.

All of your EJBs are now ready to export to EJB 1.1 JARs.

7.5.5 Export EJBs as EJB 1.1 JAR files

The next stage is to export your EJBs into EJB 1.1 compliant JAR files.

To export your EJBs into EJB 1.1 JAR files, complete the following steps:

1. Switch to the EJB view in the Workbench by clicking the **EJB** tab.
2. Right-click on the EJB group containing the EJBs to export and select **Export -> EJB 1.1 JAR...** from the menu.

The Export to an EJB 1.1 JAR File SmartGuide will appear.

3. Choose a path and file name for the JAR file and enter it into the **JAR file** text box.
4. Ensure that **DB2 for NT, V7.1** is selected in the target database list.
5. Check the options for **.class** and **.java**. Uncheck **resource** if it is selected.
6. Click **Finish** to start the export.

Even with small numbers of EJBs, this can take a long time.

Once you have exported all of your EJBs, you should move the JAR file you exported to either the system containing your installation of WebSphere Studio Application Developer V5.1.1, or into a directory that is visible from that system.

7.5.6 Import the EJBs into WebSphere Studio Application Developer

The next step is to import your EJB 1.1 JAR file into WebSphere Studio Application Developer. Once this is done, there will be some code changes to make before the EJBs will function correctly.

To import the EJB 1.1 JAR file into WebSphere Studio Application Developer V5.1.1, complete the following steps:

1. If it is not already open, launch WebSphere Studio Application Developer V5.1.1.
2. Switch to the **J2EE Perspective** by clicking its icon or clicking **Window -> Open Perspective -> J2EE**.
3. Click on the **Project Navigator** tab.
4. Right-click on the **WebSphereCommerceServerExtensionsData** project and select **Import...** from the menu.

The Import window will appear.

5. Select **EJB JAR file** from the list of sources and click **Next**.
6. Next to the **EJB JAR file** list, click **Browse...** and select the EJB 1.1 JAR file that you exported in the previous section.
7. Click on the **EJB project** list and select **WebSphereCommerceServerExtensionsData** from the list and click **Next**.
8. Ensure that **Import all enterprise beans and files** is selected and click **Finish**.

When you have imported all your EJBs there are still several more steps you need to take for the EJBs to work properly. At this point, you will notice there are errors and warnings in your EJB code. These errors will be fixed when you follow the post migration steps in 8.2.6, "Configuration and code changes for migrated EJBs" on page 144.



Post-migration steps

This chapter describes the additional steps to be performed after the development instance and database have been migrated.

This chapter is organized in the following sections:

- ▶ Setting up aliases
- ▶ Migrating custom code
 - Access control
 - Command parameter validation
 - User registration
 - Logon command
 - Calculation usage framework
 - Configuration and code changes for migrated EJBs
 - J2EE Connector Architecture
 - Pricing
 - Product Advisor
 - Rule server administration commands
 - JSP and property file changes

8.1 Setting up aliases

The application that we migrated contained static web content that was addressed outside the scope of the web application in order to minimize the load on the application server.

The WebSphere Commerce V5.6 Toolkit does not contain any static web projects when installed, so in order to support static web content, you must do the following:

1. Create a new Dynamic Web Project:
 - a. File -> New -> Dynamic Web Project
 - b. The **New Web Project** wizard appears. Ensure that **Configure advanced options** is checked, enter the name of the project in the **Project name** entry field and click **Next**.
We used the name `static_http`.
 - c. The **J2EE Setting Page** appears. Enter the prefix the **Context root** entry field that your static web content must use and click **Next**.
We needed support for all paths that are not used for WebSphere Commerce servlets, so we entered `/` (forward slash).
 - d. The Features Page appears. Ensure that none of the features are checked and click **Next**.
 - e. The **Select a Page Template for the Web Site** page appears. Ensure that **Use a default Page Template for the Web Site** is unchecked and click **Finish**.
 - f. The **New Web Project** wizard starts creating the web project. When finished, a prompt asking if you want to switch to the Web Perspective. Click **No**.
2. Edit the **looseconfig.xml** file located in one of the following directories, depending on the type of test environment that you use:

- Full test environment:

```
<wctoolkit_home>\conf\servers\fullconfig\looseconfig.xml
```

- Lightweight test environment:

```
<wctoolkit_home>\conf\servers\lightconfig\looseconfig.xml
```

As shown in the excerpt in Example 8-1. The part in bold shows the new content. If you entered a different name for the Web Project replace `static_http` in the file for the one you just entered.

Example 8-1 Excerpt from the looseconfig.xmi file

```

:
:
:
<looseArchives binariesPath="..\workspace\Stores\Web Content"
resourcesPath="..\workspace\Stores\Web Content" uri="Stores.war"
xsi:type="com.ibm.etools.commonarchive.looseconfig:LooseWARFile"/>
<looseArchives binariesPath="..\workspace\static_http\WebContent"
resourcesPath="..\workspace\static_http\WebContent" uri="static_http.war"
xsi:type="com.ibm.etools.commonarchive.looseconfig:LooseWARFile"/>
<looseArchives binariesPath="..\wars\Payments.war"
resourcesPath="..\wars\Payments.war" uri="Payments.war"
xmi:type="com.ibm.etools.commonarchive.looseconfig:LooseWARFile"/>
:
:
:

```

3. Edit the **deployment.xml** file located in one of the following directories, depending on the type of test environment that you use:

- Full test environment:

```

<wctoolkit_home>\conf\servers\fullconfig\cells\localhost\applications\wc
full.ear\deployments\wcfull\deployment.xml

```

- Lightweight test environment:

```

<wctoolkit_home>\conf\servers\lightconfig\cells\localhost\applications\w
clight.ear\deployments\wclight\deployment.xml

```

As shown in Example 8-2. If you entered a different name for the Web Project replace `static_http` in the file for the one you entered.

The value for `startingWeight` determines the starting order for modules in the server. The modules with smaller the values are started first.

The value for `xmi:id` (`WebModuleDeployment_567` in the example) should be chosen such that it is unique within the **deployment.xml** file.

Example 8-2 Excerpt from the deployment.xml file

```

:
:
:
<modules startingWeight="10010" uri="Stores.war" xmi:id="WebModuleDeployment_5"
xmi:type="appdeployment:WebModuleDeployment">
<targetMappings target="ServerTarget_1" xmi:id="DeploymentTargetMapping_18"/>
</modules>
<modules startingWeight="10020" uri="static_http.war"
xmi:id="WebModuleDeployment_567" xmi:type="appdeployment:WebModuleDeployment">
<targetMappings target="ServerTarget_1" xmi:id="DeploymentTargetMapping_567"/>
</modules>

```

```

<modules startingWeight="1010" uri="wcs.war" xmi:id="WebModuleDeployment_6"
xmi:type="appdeployment:WebModuleDeployment">
<targetMappings target="ServerTarget_1" xmi:id="DeploymentTargetMapping_19"/>
</modules>
:
:
:

```

4. Edit the **application.xml** file located in one of the following directories, depending on the type of test environment that you use:

- Full test environment:

```
<wctoolkit_home>\applications\wcfull.ear\META-INF
```

- Lightweight test environment:

```
<wctoolkit_home>\applications\wclight.ear\META-INF
```

As shown in the excerpt in Example 8-3. If you entered a different name for the Web Project replace `static_http` in the file for the one you entered.

The value for the `id` attribute on the `module` node (`WebModule_1041293846136` in the example) should be chosen such that it is unique within the **application.xml** file.

The value in the `context-root` node must match the value entered in the **Context root** entry field when you created the Web project.

Example 8-3 Excerpt from the application.xml file

```

:
:
:
<module id="WebModule_1041293846133">
  <web>
    <web-uri>Stores.war</web-uri>
    <context-root>/webapp/wcs/stores</context-root>
  </web>
</module>
<module id="WebModule_1041293846136">
  <web>
    <web-uri>static_http.war</web-uri>
    <context-root>/</context-root>
  </web>
</module>
<module id="WebModule_1041293846134">
  <web>
    <web-uri>wcs.war</web-uri>
    <context-root>/wcs</context-root>
  </web>
</module>

```

:
:
:

5. Start or restart the server in WebSphere Studio Application Developer.

8.2 Migrating custom code

This section describes the mandatory and optional changes necessary to complete the migration from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

The section is divided into the following parts:

- ▶ Access control
- ▶ Command parameter validation
- ▶ User registration
- ▶ Logon command
- ▶ Calculation usage framework
- ▶ Configuration and code changes for migrated EJBs
- ▶ J2EE Connector Architecture
- ▶ Pricing
- ▶ Product Advisor
- ▶ Rule server administration commands
- ▶ JSP and property file changes

8.2.1 Access control

The access control framework has changed significantly from WebSphere Commerce Suite V5.1.

In WebSphere Commerce Suite V5.1, high level access control was controlled using role based access control, defined by the ACCCMDGRP, ACCCUSTEXC and ACCMBRGRP tables, specifying which member groups were allowed to access which commands and views. Fine grained resource control would be defined programmatically in WebSphere Commerce Suite V5.1.

In WebSphere Commerce V5.6 the access control framework has changed to be policy based. This means that commands, views and data beans can be grouped into resource groups for which access control can be specified using access control policies. Access control in WebSphere Commerce V5.6 is typically configured using the `acpload` utility. Refer to the WebSphere Commerce InfoCenter for details of implementing access control.

In WebSphere Commerce Suite V5.1, fine-grained resource control was implemented programatically using the following methods:

```
public Long[] getResourceOwners() throws ECEException  
public boolean checkPermission() throws ECEException
```

These methods became deprecated in WebSphere Commerce Suite V5.4. Instead the fine-grained resource control has become policy based, combined with the use of the following methods:

```
public AccessVector getResources() throws ECEException  
public void checkResourcePermission() throws ECEException
```

Figure 8-1 on page 131 shows the method call sequence for invoking commands in the WebSphere Commerce V5.6 command framework.

Note: Refer to the WebSphere Commerce InfoCenter for more details at the following URL:

<http://publib.boulder.ibm.com/infocenter/wc56help>

8.2.2 Command parameter validation

The following method was used in controller, view and task commands in WebSphere Commerce Suite V5.1 to perform server side validation of the parameters passed to a command:

```
protected void checkParameters() throws ECEException
```

This method was replaced in WebSphere Commerce Suite V5.4 with the following method, deprecating the use of the `checkParameters` method:

```
public void validateParameters() throws ECEException
```

The method is used for the same purpose with the only difference that where WebSphere Commerce Suite V5.1, the `checkParameters` method was called from within the `performExecute` method, the `validateParameters` method is called from the command framework.

Since `validateParameters` is called from the command framework, you should not replace calls in the `performExecute` method to the `checkParameters` method with calls to `validateParameters`. The custom code should thus not contain any references to its own `validateParameters` method.

Figure 8-1 on page 131 shows the method call sequence for invoking commands in the WebSphere Commerce V5.6 command framework.

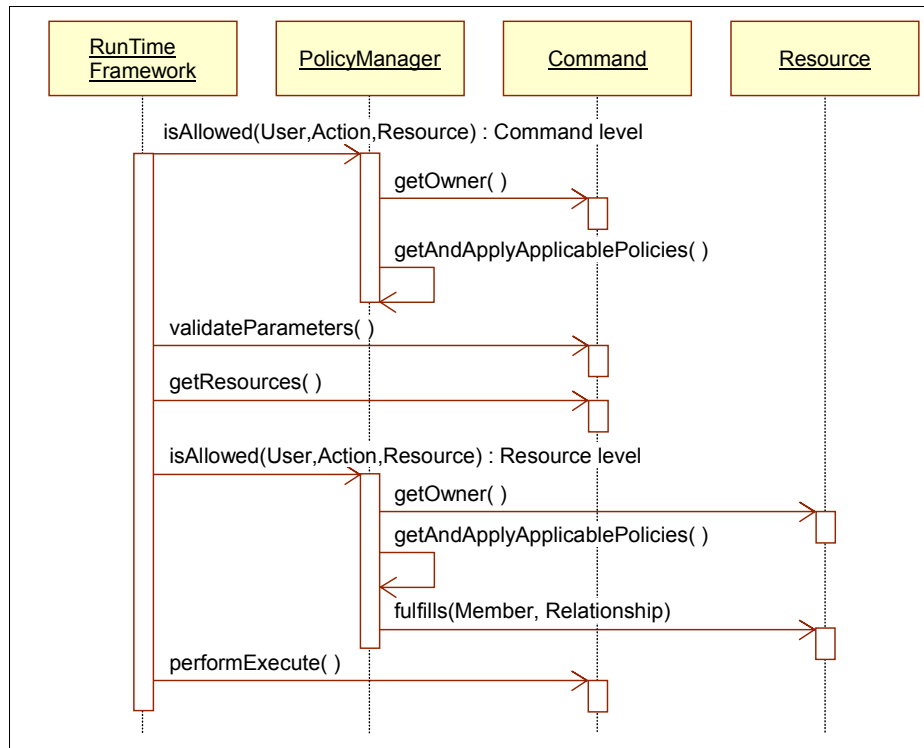


Figure 8-1 Sequence diagram for WebSphere Commerce V5.6 command framework

8.2.3 User registration

The following WebSphere Commerce Suite V5.1 command was removed in WebSphere Commerce Suite V5.4

```
com.ibm.commerce.usermanagement.commands.UserRegistrationFormCmd
```

When executed the command did the following:

1. Check if the `userId` parameter, if supplied, match the ID of the current user.
2. Invoke the command

```
com.ibm.commerce.usermanagement.commands.PostUserRegistrationFormCmd
```

The default implementation for this command does nothing.

3. Direct the user to the view `UserRegistrationForm`.

If your WebSphere Commerce Suite V5.1 application makes use of the `UserRegistrationFormCmd` command, you must either replace the command with a custom command or forward directly to the `UserRegistrationForm` view. The

latter approach is recommended, as validation of the user ID should happen in the update command.

Important: UserRegistrationUpdateCmd, UserRegistrationAddCmd and UserRegistrationDeleteCmd must be invoked as URL commands.

If these commands are invoked from a JSP called directly through a view, or from a generic controller command (a command that returns true from the isGeneric method), the commands may update or delete the generic user (the user with ID -1002). This can have adverse effects on the behavior of the entire site.

8.2.4 Logon command

The behavior of the logon command is different in WebSphere Commerce V5.6, compared to WebSphere Commerce Suite V5.1. The following restrictions have been added to users attempting to log on to a store in WebSphere Commerce V5.6:

- ▶ The user must have a role in the store's organization hierarchy.
- ▶ The user's registration has to be approved. If the user is pending approval, he or she will not be able to log on.
- ▶ If any of the organizations in the user's ancestral organization hierarchy is locked, the user will not be allowed to log on.

While these changes do not necessarily prompt any code changes in custom code, they may affect the default behavior of your store.

8.2.5 Calculation usage framework

If you have made customization in the calculation framework, these should be migrated to the new structure in WebSphere Commerce V5.6.

Note: It is recommended that you familiarize yourself with the information about the calculation framework, found in the WebSphere Commerce V5.6 infoCenter at the following address:

<http://publib.boulder.ibm.com/infocenter/wc56help>

Background

In WebSphere Commerce Suite V5.1, new logic for the calculation framework could be implemented by replacing the implementation class for one of the following task commands, depending on the calculation type:

- ▶ `com.ibm.commerce.order.utils.ApplyCalculationUsagesCmd`
- ▶ `com.ibm.commerce.price.commands.ApplyOrderAdjustmentsCmd`
- ▶ `com.ibm.commerce.fulfillment.commands.ApplyOrderShippingChargesCmd`
- ▶ `com.ibm.commerce.taxation.commands.ApplyOrderTaxesCmd`

Typically, the implementation classes for these commands would be referenced by the custom implementation, either by extending or other means. If this is the case, the custom code will have to be modified to work in WebSphere Commerce V5.6. Even if the custom code does not reference the implementation classes it is recommended to migrate to the new calculation framework.

In WebSphere Commerce V5.6, customization to the calculation framework should be done by replacing one of the following task commands:

- ▶ `com.ibm.commerce.order.calculation.ApplyCalculationUsageCmd`
- ▶ `com.ibm.commerce.order.calculation.ApplyShippingCmd`

Note: Placed in different packages but named very similar, the WebSphere Commerce Suite V5.1 command `ApplyCalculationUsagesCmd` differs from the WebSphere Commerce V5.6 command `ApplyCalculationUsageCmd`.

Table 8-1 shows the mapping between the WebSphere Commerce Suite V5.1 and WebSphere Commerce V5.6 commands for the different calculation usages.

Table 8-1 Mapping between WebSphere Commerce commands

ID	Calculation usage	WebSphere Commerce Suite V5.1 command	WebSphere Commerce V5.6 command
-1	Discount	<code>ApplyOrderAdjustmentsCmd</code>	<code>ApplyCalculationUsageCmd</code>
-2	Shipping	<code>ApplyOrderShippingChargesCmd</code>	<code>ApplyShippingCmd</code>
-3	Sales Tax	<code>ApplyOrderTaxesCmd</code>	<code>ApplyCalculationUsageCmd</code>
-4	Shipping Tax	<code>ApplyOrderTaxesCmd</code>	<code>ApplyCalculationUsageCmd</code>

Note: Table 8-1 does not show any mapping for the WebSphere Commerce Suite V5.1 ApplyCalculationUsagesCmd command. The reason for this is that this command is not meant to be used independently. Instead, this command is the base class for the rest of the calculation commands mentioned.

Additionally, Table 8-1 does not display the calculate usage for coupons as this was not available in WebSphere Commerce Suite V5.1.

It should also be noted that the default WebSphere Commerce V5.6 commands may change in the future. To verify the current command classes for the various calculation usages, execute the following SQL statement:

```
select calusage_id, taskname
  from calmethod
  where storeent_id=-1
  and subclass=12
  order by calusage_id desc
```

Migration

If any of the WebSphere Commerce Suite V5.1 commands of Table 8-1 on page 133 are being referenced or otherwise customized, these must be migrated to use the corresponding WebSphere Commerce V5.6 command.

You can determine if any of these commands have been extended in your WebSphere Commerce Suite V5.1 store by issuing the following SQL:

```
select * from cmdreg where interfacename like '%.Apply%'
```

If this SQL returns any rows with the INTERFACENAME column equal to one of the WebSphere Commerce Suite V5.1 commands mentioned in “Background” on page 132, you must migrate these as described in this section.

The following instructions show how to migrate a command that extends one of the WebSphere Commerce Suite V5.1 commands into a command that extends the corresponding WebSphere Commerce V5.6 command:

1. Remove the entry in the command registry by issuing the following SQL:

```
delete from cmdreg interfacename='<wc51interfacename>'
```

Where <wc51interfacename > is name of the WebSphere Commerce Suite V5.1 command interface name from Table 8-1 on page 133.

In our example, we had a customized version of the ApplyOrderShippingChargesCmd command, so we issued the following SQL:

```
delete from cmdreg where
interfacename='com.ibm.commerce.fulfillment.commands.ApplyOrderShippingC
hargesCmd'
```

2. Use Table 8-1 on page 133 to identify the WebSphere Commerce V5.6 interface that match the WebSphere Commerce Suite V5.1 interface used in the previous step.

In our example, the WebSphere Commerce V5.6 interface is
`com.ibm.commerce.order.calculation.ApplyShippingCmd`.

3. Create a custom interface, extending the the WebSphere Commerce V5.6 command interface. Add two constants, `NAME` and `defaultCommandClassName` to the interface, containing the name, including package, of the custom interface and the implementing class, respectively. The implementing class is created in the next step.

Example 8-4 Sample MyApplyShippingCmd

```
package mypackage;

public interface MyApplyShippingCmd
    extends com.ibm.commerce.order.calculation.ApplyShippingCmd
{
    public static final String NAME = "mypackage.MyApplyShippingCmd";
    public static final String defaultCommandClassName =
        "mypackage.MyApplyShippingCmdImpl";
}
```

4. Create a custom class, implementing the interface created in the previous step and extending from the WebSphere Commerce V5.6 command implementation class. Refer to “Interface changes” on page 139 for details about the interface changes from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

Our resulting implementation class is shown in Example 8-5.

Example 8-5 Sample MyApplyShippingCmdImpl

```
package mypackage;

import java.util.Set;
import java.util.HashSet;

public class MyApplyShippingCmdImpl
    extends com.ibm.commerce.order.calculation.ApplyShippingCmdImpl
    implements MyApplyShippingCmd
{
    private static final long COMPONENT = ECTraceIdentifiers.COMPONENT_EXTERN;
    private Item[] iItems = new Item[0];
    private Set iAppliedItems = new HashSet();

    /**
     * Get the list of order items that must be processed.
```

```
    */
    protected Item[] getItems()
    {
        return iItems;
    }

    /**
     * Set the list of order items that must be processed.
     */
    public void setItems(Item[] items)
    {
        super.setItems(items);
        iItems = items;
    }

    /**
     * Apply order adjustments to all items in iItems.
     * Update iAppliedItems with the items that have had order adjustments
     * applied to them.
     */
    public void performExecute()
    {
        :
        :
        // customized shipping calculation...
        // when the shipping charge for an order item has been
        // applied, the corresponding object from the iItems array
        // will be added to the iAppliedItems set for validation in
        // the checkAppliedItems method.
        :
        :
    }

    /**
     * Return a Set containing instances of the Items that have had
     * applied an order adjustment to them.
     */
    public Set getAppliedItems()
    {
        return iAppliedItems;
    }

    /**
     * Checks if a value has been calculated and applied to each and
     * every item.
     * If not, an ECAApplicationException with ECMessages _ERR_CALC_CODE will
     * be thrown.
     * @exception ECAException
     */
```

```

public void checkAppliedItems()
    throws ECEException
{
    final String METHOD_NAME = "checkAppliedItems";

    if(ECTrace.traceEnabled(COMPONENT))
    {
        ECTrace.trace(
            COMPONENT, CLASS_NAME, METHOD_NAME,
            "itemIds=" + CalculationHelper.getInstance().toString(iItems));
        ECTrace.trace(
            COMPONENT, CLASS_NAME, METHOD_NAME,
            "appliedItems=" + appliedItems);
    }

    for(int i = 0; i < iItems.length; i++)
    {
        Set appliedItems = getAppliedItems();
        if(appliedItems == null || !appliedItems.contains(iItems[i]))
        {
            TypedProperty exceptionData = iItems[i].createExceptionData();

            exceptionData.put(
                ECConstants.EC_ERROR_CODE,
                CalculationConstants.ERRCODE_SHIPPING_NOT_APPLIED);
            exceptionData.put(
                CalculationConstants.EC_USAGE_ID,
                "" + getUsageId());

            throw new ECApplException(
                ECMessage._ERR_CALCODE, CLASS_NAME, METHOD_NAME,
                new Object[] {
                    getUsageId().toString(),
                    CalculationConstants.ERRCODE_SHIPPING_NOT_APPLIED },
                CalculationConstants.CALCODE_ERRVIEW, exceptionData);
        }
    }
}
}
}

```

-
5. Optional: Register the new command implementation in the command registry using the following SQL:

```

insert into cmdreg
    (storeent_id, interfacename, classname, lastupdate)
values

```

```
(<storeId>, <custominterface>, <customclass>, current timestamp)
```

Where <storeId> is the store ID, <wc56interfacename> is the name of the WebSphere Commerce V5.6 interface, matching the original WebSphere Commerce Suite V5.1 interface, and <customclass> is the full name, including package, of the custom version of the WebSphere Commerce V5.6 implementation class.

In our example, we used the following SQL:

```
insert into cmdreg
(storeent_id, interfacename, classname, lastupdate)
values
(10001, 'mypackage.MyApplyShippingCmd',
'mypackage.MyApplyShippingCmdImpl', current timestamp)
```

6. Register the new command as a calculation method in the calculation method registry by issuing the following SQL:

```
insert into calmethod (calmethod_id, storeent_id, calusage_id, taskname,
subclass)
values (<newMethodId>, <storeId>, <calusageId>, <custominterface>, 12)
```

Where <newMethodId> is a new unique value for the CALMETHOD_ID column of the CALMETHOD table, <storeId> is the ID of the store, <calusageId> is the calculation usage ID for the new command (refer to Table 8-1 on page 133), and <custominterface> is the full name, including package, of the interface created earlier.

We used the following SQL:

```
insert into calmethod (calmethod_id, storeent_id, calusage_id, taskname,
subclass)
values (1000, 10001, -2, 'mypackage.MyApplyShippingCmd', 12)
```

7. Register the new calculation method to be used in your store for applying the corresponding calculation usage. This is done by adding an entry to the STENCALUSG table. This table will already contain entries for the default methods, using the store ID -1. As such, you can base the entry for your store on one of these entries shown in the following example:

Example 8-6 Registering calculation method

```
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)

select <storeId>, calusage_id, actcc_calmethod_id, actrc_calmethod_id,
calcode_id, <newMethodId>, calmethod_id_sum, calmethod_id_fin, usageflags,
calmethod_id_ini, sequence
from stencalusg
where calusage_id=<calusageId>
```

and storeent_id=-1

Where <storeId> is the ID of the store, <newMethodId> is the ID of the new entry that was added to the CALMETHOD table and <calusagId> is the calculation usage ID from Table 8-1 on page 133.

Interface changes

This section gives an overview of the most important changes from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6 in the signature of the payment commands.

This is to ease the migration of commands that extend or otherwise reference the existing payment commands in WebSphere Commerce Suite V5.1.

WebSphere Commerce V5.6 command interfaces

ApplyCalculationUsageCmd and its subinterface ApplyShippingCmd have the following methods, relevant to developers that wish to extend the built-in implementation classes ApplyCalculationUsageCmdImpl and ApplyShippingCmdImpl:

- ▶ `public void setCurrency(String)`

This method replaces the WebSphere Commerce Suite V5.1 `setOrderCurrency` method. This method is only described here as it replaces a used WebSphere Commerce Suite V5.1 method. Classes extending from the `ApplyCalculationUsageCmdImpl` and `ApplyShippingCmdImpl` classes should use the `getCurrency` method to retrieve the order currency.

- ▶ `public String getCurrency()`

This method can be used by classes that extend `ApplyCalculationUsageCmdImpl` or `ApplyShippingCmdImpl` to determine the order currency. The introduction of this method to the command interface for the calculation framework means WebSphere Commerce Suite V5.1 code that overrides the `setOrderCurrency` method can be eliminated, as this was typically done to retrieve the order currency.

- ▶ `public void checkAppliedItems()`

This method is called by the framework for commands that are registered in the store calculation usage registry, STENCALUSG, as being required to apply adjustments to all order items. This is done by the check bit flag in the USAGEFLAGS column. Refer to the database documentation in the infoCenter for details:

<http://publib.boulder.ibm.com/infocenter/wc56help>

- ▶ `public java.util.Set getAppliedItems()`

This method should return a `java.util.Set` instance containing the `Item` instances that have had adjustments calculated and applied. The method should be used in the `checkAppliedItems` method.

Note: We found that at the time of writing, the `ApplyShippingCmdImpl` and `ApplyCalculationUsageCmdImpl` command implementations did not use the `getAppliedItems` method from the `checkAppliedItems` method. This means that if you extend any of these commands with a calculation command that has the check bit flag set in `STENCALUSG.USAGEFLAGS`, you must also provide your own implementation of the `checkAppliedItems` method.

See Example 8-5 on page 135 for a sample implementation of the method `checkAppliedItems`.

- ▶ `public void setItems(Item[])`

This method is used by the framework to pass an array of `Item` instances, representing the order items that should be processed, corresponding to the WebSphere Commerce Suite V5.1 `setOrderItems` methods.

If you extend the `ApplyCalculationUsageCmdImpl` class, you do not need to implement this method, as this class contains the following method that can be used to retrieve the list of items that should be processed:

```
public Item[] getItems()
```

If you extend the `ApplyShippingCmdImpl` class, you will have to override the `setItems` method and store the array locally before calling `super.setItems`, as the `getItem` method in `ApplyShippingCmdImpl` has the following signature:

```
private Item[] getItems()
```

Refer to Example 8-5 on page 135 for an sample of this.

- ▶ `public void setTaxCategoryIds(Integer[])`

This method is unchanged from WebSphere Commerce Suite V5.1 and is used

WebSphere Commerce Suite V5.1 command interfaces

The WebSphere Commerce Suite V5.1 interfaces for the commands `ApplyOrderShippingChargesCmd`, `ApplyOrderTaxesCmd` and `ApplyOrderAdjustmentsCmd` contain the following methods:

- ▶ `public void setOrderCurrency(String)`

This method has been made obsolete by the introduction of the `getCurrency` method in the `CalculationCmd` interface, which is the ancestor interface to `ApplyCalculationUsageCmd` and `ApplyShippingCmd` interfaces.

► `public void setOrderItems(...)`

This method existed in two overloaded versions, taking either an array or a `java.util.Enumeration` of `OrderItemAccessBean` instances.

The use of this method has been replaced by the use of the `setItems` and `getItems` methods. Refer to “WebSphere Commerce V5.6 command interfaces” on page 139 for details.

► `public void setTaxCategoryIds(Integer[])`

This method, which was not present in the `ApplyOrderAdjustmentsCmd` interface, is unchanged from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

Accessing the order items

In WebSphere Commerce Suite V5.1, the order items was accessed directly through the `OrderItem` EJB’s access bean, `com.ibm.commerce.order.objects.OrderItemAccessBean`. In WebSphere Commerce V5.6 an envelope class, `com.ibm.commerce.order.calculation.Item`, has been introduced. Code that accesses the `OrderItemAccessBean` directly should be converted to use the `Item` class instead, if possible.

Table 8-2 outlines the changes that should be made. The first column list the WebSphere Commerce Suite V5.1 code that have corresponding WebSphere Commerce V5.6 code, accessing the `Item` class instead. If access to the `OrderItemAccessBean` class cannot be avoided, `Item` has an accessor method to retrieve a reference to the underlying `OrderItemAccessBean`, as shown in the first row of Table 8-2.

Tip: Mapping OrderItemAccessBean to Item considerations:

- If an accessor is not available in the interface to Item, it is possible to access the underlying OrderItemAccessBean using the following method:

```
public OrderItemAccessBean getOrderItem()
```

This should be avoided, if at all possible.

- Table 8-2 shows only the accessors that directly map between OrderItemAccessBean and Item. If, for example, the WebSphere Commerce Suite V5.1 code contains the line:

```
String strQuantity = oiab.getQuantity()
```

Then this could be migrated into the following line:

```
String strQuantity = item.getOrderItem().getQuantity()
```

But most likely, the code should be rewritten to use the `getQuantity` method of the Item class, using `BigDecimal` objects instead strings to hold the order quantity.

- If any of the setter methods on the Item class are used, the `commit` method should be used to commit the changes to the underlying OrderItemAccessBean:

```
item.setShippingTaxTotal(bdTaxTotal);
item.setShippingTotal(bdTotal);
item.commit();
```

- If it cannot be avoided to modify values directly in the OrderItemAccessBean, the envelope Item bean should be refreshed in addition calling `commitCopyHelper`:

```
OrderItemAccessBean oiab = item.getOrderItem();
oiab.setShippingTaxTotal("22");
oiab.setShippingTotal("220");
oiab.commitCopyHelper();
item.refresh();
```

Table 8-2 Mapping from use of the OrderItemAccessBean to the Item envelope class

Accessing OrderItemAccessBean	Accessing Item
getStoreIdInEJBType() setStoreId(1StoreId)	getStoreEntityId() setStoreEntityId(1StoreId)
getStatus() setStatus()	getStatus() setStatus()
getCatalogEntryIdInEJBType() setCatalogEntryId(1CatEntId)	getCatalogEntryId() setCatalogEntryId(1CatEntId)

Accessing OrderItemAccessBean	Accessing Item
getContractIdInEJBType() setContractId(lContractId)	getContractId() setContractId(lContractId)
getOfferIdInEJBType() setOfferId(lOfferId)	getOfferId() setOfferId(lOfferId)
getAddressIdInEJBType() setAddressId(lAddressId)	getAddressId() setAddressId(lAddressId)
getFulfillmentCenterIdInEJBType() setFulfillmentCenterId(lFfmCenterId)	getFulfillmentCenterId() setFulfillmentCenterId(lFfmCenterId)
getShippingModeIdInEJBType() setShippingModeId(iShipmode)	getShippingModeId() setShippingModeId(iShipmode)
getPrepareFlagsInEJBType() setPrepareFlags(iPrepareFlags)	getPrepareFlags() setPrepareFlags(iPrepareFlags)
getPriceInEJBType() setPrice(bdPrice)	getPrice() setPrice(bdPrice)
getQuantityInEJBType() [Double] setQuantity(dQuant)	getQuantity() [BigDecimal] setQuantity(new BigDecimal(dQuant))
getTotalProductInEJBType() setTotalProduct(bdTotal)	getProductTotal() setProductTotal(bdTotal)
getTotalAdjustmentInEJBType() setTotalAdjustment(bdTotal)	getAdjustmentTotal() setAdjustmentTotal(bdTotal)
getShippingChargeInEJBType() setShippingCharge(bdTotal)	getShippingTotal() setShippingTotal(bdTotal)
getTaxAmountInEJBType() setTaxAmount(bdTotal)	getSalesTaxTotal() setSalesTaxTotal(bdTotal)
getShippingTaxAmountInEJBType() setShippingTaxTotal(bdTotal)	getShippingTaxTotal() setShippingTaxTotal(bdTotal)
getEstimatedAvailableTimeInEJBType() setEstimatedAvailableTime(tsTime)	getEstimatedAvailableTime() setEstimatedAvailableTime(tsTime)
getShippingOffsetInEJBType() setShippingOffset(iShipOffset)	getShippingOffset() setShippingOffset(iShipOffset)
getCategoryTaxAmount(iTaxCatId) setCategoryTaxAmount(iTaxCatId,dVal)	getTaxAmount(iTaxCatId) setTaxAmount(iTaxCatId,dVal)

8.2.6 Configuration and code changes for migrated EJBs

EJBs developed in VisualAge for Java V3.5.3 follow the EJB 1.0 specification. these must be converted to EJB 1.1 specification level.

Since the process of converting involves exporting the EJB code from VisualAge for Java V3.5.3, importing the EJBs into VisualAge for Java V4.0, exporting from VisualAge for Java V4.0 and then importing into WebSphere Studio Application Developer V5.1.1, we recommend that you recreate the EJBs in WebSphere Studio Application Developer V5.1.1, unless you have a very large number of EJBs, justifying the cumbersome conversion process.

This section details the steps you must take to complete the migration of your custom EJBs. This includes the following configuration changes:

- ▶ Change the access isolation level
- ▶ Add the WCTSecurity role
- ▶ Change the container transaction type

As well as the following code changes:

- ▶ Remove serialVersionUID constants
- ▶ Ensure that ejbCreate returns primary key object
- ▶ Remove java.rmi.RemoteException from remote methods
- ▶ Match ejbPostCreate and ejbCreate methods
- ▶ Remove FinderHelper interface
- ▶ Regenerating the deploy code

It is assumed that you have already migrated your EJBs to WebSphere Studio Application Developer V5.1.1 using the process described in 7.5, “Migrating custom EJBs” on page 118.

Access bean string converters

In VisualAge for Java V3.5.3, you were able to specify string converters for the fields in the EJB when generating the access bean for the EJB, as shown in Figure 8-2 on page 145.

A similar step does not exist in the access bean editor in WebSphere Studio Application Developer V5.1.1. However, the access bean creation engine in WebSphere Studio Application Developer V5.1.1 does create the string converters for you, if you make sure that your EJBs are configured to use string converters.

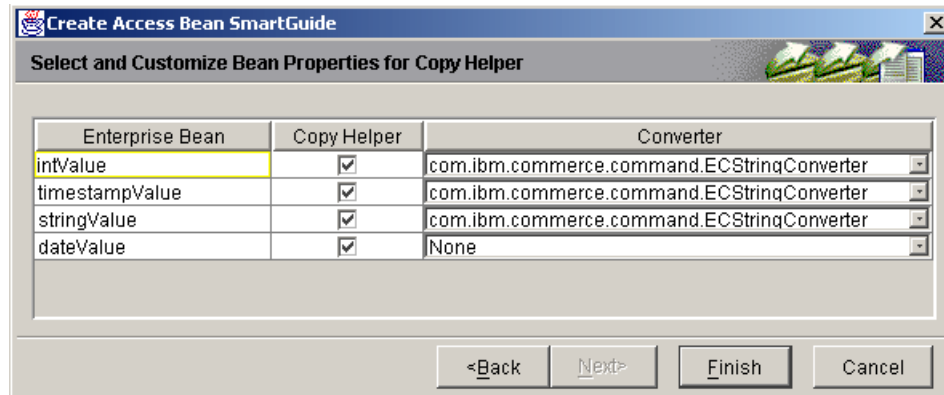


Figure 8-2 Adding string converters to an access bean in VisualAge for Java V3.5.3

The access bean configuration data is stored in the file **ibm-ejb-access-bean.xml**, placed in the `ejbModule/META-INF` folder of the `WebSphereCommerceServerExtensionsData` project in WebSphere Studio Application Developer V5.1.1. A sample file is shown in Example 8-7. Note that `field1` and `field2` have set up string converters, whereas `field3` does not. As `field1` is already of type `java.lang.String`, no converters will be added to the access bean, but as `field2` is of type `java.lang.Long`, string conversion accessors will be generated.

The access bean for the EJB shown in Example 8-7 will have the following field accessors:

- ▶ `public java.lang.String getField1()`
- ▶ `public void setField1(java.lang.String)`
- ▶ `public java.lang.String getField2()`
- ▶ `public java.lang.Long getField2InEJBType()`
- ▶ `public void setField2(java.lang.String)`
- ▶ `public void setField2(java.lang.Long)`
- ▶ `public java.sql.Date getField3()`
- ▶ `public void getField3(java.sql.Date)`

The **ibm-ejb-access-bean.xml** file will automatically be configured with the string converters that have been set up for the EJB's access bean in VisualAge for Java V3.5.3 when the EJB 1.1 JAR file is imported into WebSphere Studio Application Developer. The only time you will need to edit this file manually is if you choose to recreate your EJBs manually, or if you need to add fields to an existing access bean and need to add string converters to the accessor methods for the new field.

Example 8-7 Example ibm-ejb-access-bean.xmi file

```

<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:accessbean="accessbean.xmi" xmlns:ejb="ejb.xmi">
  <accessbean:EJBShadow xmi:id="EJBShadow_1086361148891"
    name="Test" factoryPackageName="com.mycompany.ejb">
    <accessBeans xmi:type="accessbean:Type2AccessBean"
      xmi:id="Type2AccessBean_1086361148891" name="TestAccessBean"
      package="com.mycompany.ejb" version="WSAD-1.0">
      <copyHelperProperties xmi:id="CopyHelperProperty_1086361148891"
        name="field1" type="java.lang.String"
        getterName="getField1"
        setterName="setField1"
        converterClassName="com.ibm.commerce.command.ECStringConverter"/>
      <copyHelperProperties xmi:id="CopyHelperProperty_1086361148892"
        name="field3" type="java.sql.Date"
        getterName="getField3"
        setterName="setField3"/>
      <copyHelperProperties xmi:id="CopyHelperProperty_1086361148893"
        name="field2" type="java.lang.Long"
        getterName="getField2"
        setterName="setField2"
        converterClassName="com.ibm.commerce.command.ECStringConverter"/>
      <nullConstructor xmi:id="MethodElement_1086361148891"
        name="create" parms="java.lang.Long" type="Home">
        <enterpriseBean xmi:type="ejb:ContainerManagedEntity"
          href="META-INF/ejb-jar.xml#Test"/>
      </nullConstructor>
      <nullConstructorParameters xmi:id="Parameter_1086361148891"
        name="id" type="java.lang.Long" isFieldFromKey="false"/>
    </accessBeans>
    <enterpriseBean xmi:type="ejb:ContainerManagedEntity"
      href="META-INF/ejb-jar.xml#Test"/>
  </accessbean:EJBShadow>
</xmi:XMI>

```

Change the access isolation level

EJBs built for WebSphere Commerce Suite V5.1 were expected to have their isolation level set the Read Committed. For WebSphere Commerce V5.5 and above, this was changed to Repeatable Read being the required value. In order to comply with this, your migrated beans must have their isolation levels updated. To do this, complete the following steps:

1. If it is not already open, launch WebSphere Studio Application Developer V5.1.1.

2. Switch to the **J2EE Perspective** by clicking its icon or clicking **Window -> Open Perspective -> J2EE**.
3. Expand **WebSphereCommerceServerExtensionsData** and open the **EJB Deployment Descriptor**.
4. Click on the **Access** tab.
5. In the Isolation Level section:
 - a. Click **Committed** and then click **Remove**.
 - b. Click **Add....**
 - c. Ensure **Repeatable read** is selected and click **Next**.
 - d. Select all of your custom EJBs and click **Next**.
 - e. Expand each of your EJBs in turn and select the entry for all Home methods. That is the blue method icon with the superscript *H* next to it and the asterisk as the name. This should be the first entry on the list. See Figure 8-3.

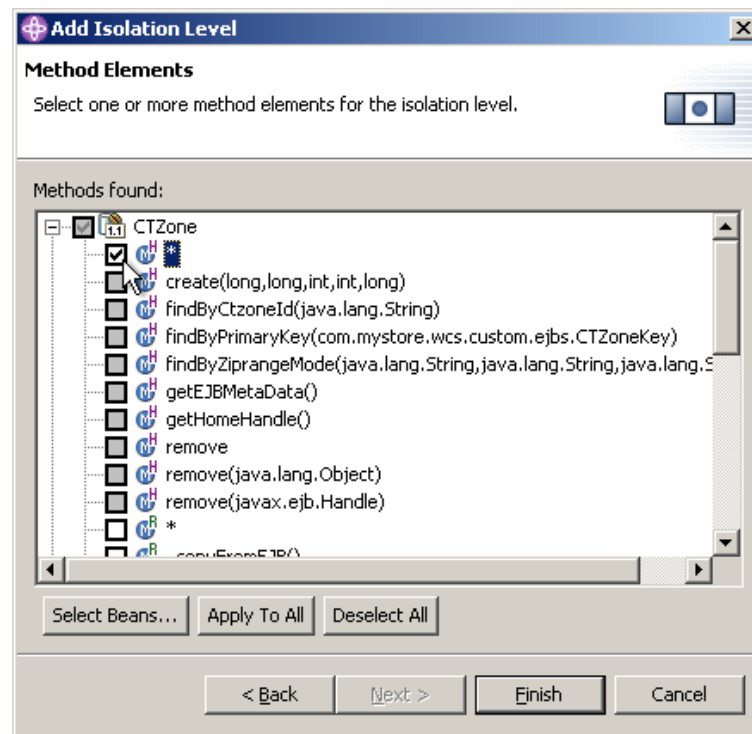


Figure 8-3 Adding the isolation level to custom EJBs

- f. Click **Finish**.

- g. Save your changes by pressing **Ctrl + S** and close the EJB Deployment Descriptor document.

Remember, these changes will not take effect until you have regenerated the deploy code for your EJBs in “Regenerating the deploy code” on page 152.

Add the WCSSecurity role

Follow these instructions to add the WCSSecurity role to all methods in your beans.

Tip: The following steps can be performed individually for each EJB, but as the steps apply to all custom EJBs, used in WebSphere Commerce V5.6, we recommend that you use the tips mentioned in the following to apply the changes to all EJBs.

1. Switch to the **J2EE Perspective** by clicking its icon or clicking **Window -> Open Perspective -> J2EE**.
2. Expand **WebSphereCommerceServerExtensionsData** and open the **EJB Deployment Descriptor**.
3. Click on the **Assembly Descriptor** tab.
4. If **WCSSecurityRole** is not present in the list box in the **Security Roles** section, do the following:
 - a. Click the **Add...** button in the **Security Roles** section.
 - b. The Add Security Role window will open. Enter **WCSSecurityRole** in the **Name** entry field and click **Finish**.
5. Click the **Add...** button in the **Method Permissions** section.
6. In the **Add Method Permissions** window, check **WCSSecurityRole** and click **Next**.
7. In the **Enterprise Beans Selection** window, check your EJBs and click **Next**.

Tip: To select all the beans, click the **Select All** button to the right.

8. In the **Method Elements** window, check each EJB that needs to be updated and click **Finish**.

Tip: To select all methods in all beans, click the **Apply To All** button below the bean list.

9. Press **Ctrl + S** to save your changes

Change the container transaction type

From the **Assembly Descriptor** tab of the **EJB Deployment Descriptor**. (see , “Add the WCSSecurity role” on page 148 for details), do the following:

1. If any transaction levels are already associated with your EJBs, you may wish to remove them and re-add them again. This is easier than finding which need to be updated and which do not.
2. In the Container Transactions section, Click **Add....**
The Add Container Transaction window will open.
3. Select your custom EJBs and click **Next**.
4. Click the **Container transaction type** list and select **Required**.
5. In the Methods found box, select all of your custom EJBs and click **Finish**.
6. Save your work by pressing **Ctrl + S**.

Note: If you have any EJBs which required the transaction level to be RequiresNew, repeat the steps above for those EJBs. If you are unsure about this, you can check the settings in the backup of your VisualAge for Java environment.

Remove serialVersionUID constants

Although this is not a strict requirement of the EJB 1.1 specification, you will need to remove any constant declarations of the name serialVersionUID from your Bean source.

Open each of your Bean Java source files in turn. In the class level variable declarations section, you will see a line of code similar to the following:

```
public final static String serialVersionUID = <unique identifier>
```

Delete this line and save your source by pressing **Ctrl + S**.

Ensure that ejbCreate returns primary key object

For entity beans, the ejbCreate methods must return the primary key object that correspond to the object that the method has created.

This is an EJB 1.1 specification requirement. For details about the requirements for ejbCreate method on entity beans, see sections 9.2.3, 9.4.2 and 9.4.7.3 of the EJB 1.1 specification, which can be found at:

<http://java.sun.com/products/ejb/docs.html>

Note: As the `ejbCreate` method on session beans must have return type `void`, they are not encompassed by this change. See section 6.10.3 of the EJB 1.1 specification for details on `ejbCreate` methods for session beans.

If one of the `ejbCreate` methods in your entity bean is not declared to return an object of the primary key type, WebSphere Studio Application Developer V5.1.1 will show the following warning in the Tasks panel:

CHKJ2406W: The method should return the primary key type <BeanName>Key.
(EJB 1.1: 9.2.3,9.4.2,9.4.7.3)

Where <BeanName> is the name of the bean, including the java package.

Remove `java.rmi.RemoteException` from remote methods

As per EJB 1.1 specification, throwing `java.rmi.RemoteException` from the following methods have been deprecated:

- ▶ `ejbCreate`
- ▶ `ejbPostCreate`
- ▶ `ejbActivate`
- ▶ `ejbPassivate`
- ▶ `ejbLoad`
- ▶ `ejbStore`
- ▶ `ejbRemove`
- ▶ All remote methods

The exception `java.rmi.RemoteException` was used in EJBs following the EJB 1.0 specification to indicate non-application exceptions.

If a non-application exception needs to be signalled to the EJB container, the remote method should throw the `javax.ejb.EJBException` or another unchecked exception. An unchecked exception is an exception that derives from `java.lang.RuntimeException` and that does not need to be declared in a `throws` clause for the method.

Note: This change only pertains to the use of `java.rmi.RemoteException`. Any application specific exceptions may be declared for, and thrown in, the mentioned methods.

For some of the mentioned methods, if they have `java.rmi.RemoteException` in their `throws` clause, WebSphere Studio Application Developer V5.1.1 will show the following warning for the method in the Tasks panel:

CHKJ2400W: Deprecated use of a `java.rmi.RemoteException` (EJB 1.1: x.y.z)

Where `x.y.z` denotes the subsection number in the EJB 1.1 specification in which the use of `java.rmi.RemoteException` deprecation for the relevant method is mentioned. For `ejbCreate` and `ejbPostCreate` this will be 9.2.3 and 9.2.4, respectively.

Other methods, such as `ejbLoad`, violating this requirement will result in the following error in the Tasks panel:

```
Exception RemoteException is not compatible with throws clause in  
EJEntityBean.<methodName>
```

Where `<methodName>` is the name of the violating method. This is caused by the fact that EJBs created for use with WebSphere Commerce should extend from the `EJEntityBean` class. This class already defines some of the EJB methods without the `java.rmi.RemoteException` in the throws clause. If the extending class then declares the `java.rmi.RemoteException` in its throws clause, it broadens the signature of the method, which is not allowed in Java.

Match `ejbPostCreate` and `ejbCreate` methods

Both EJB 1.0 and 1.1 specifications require that each `ejbCreate` method has a matching `ejbPostCreate` method. However, VisualAge for Java V3.5.3 did not enforce this requirement. After importing your EJBs into WebSphere Studio Application Developer V5.1.1, you must ensure that this requirement is fulfilled for all of your entity beans.

If an `ejbCreate` method exists without a matching `ejbPostCreate` method, WebSphere Studio Application Developer V5.1.1 will show the following warning for the `ejbCreate` method in the Tasks panel:

```
CHKJ2002W: This class should implement a matching ejbPostCreate method for  
this method (EJB 1.1: 9.2.4)
```

Remove `FinderHelper` interface

The `FinderHelper` interface was used in VisualAge for Java V3.5.3 to ease the generation of finder methods for entity beans. In order to generate code to find an object, or a list of objects from an SQL WHERE clause, it was sufficient to add a constant to the `FinderHelper` interface and a method to the home interface and generate the access bean and deployed code.

For example, to generate a finder method that returns a `java.util.Enumeration` of the objects for which the `VALUE` column, assumed to be of type `INTEGER`, has a value greater than the one supplied as a parameter to the finder, the following field would be added to the `FinderHelper` interface:

```
public static final String findGreaterThanWhereClause = "T1.VALUE > ?";
```

And the following method would be added to the home interface:

```
public abstract java.util Enumeration findGreaterThan(Integer value)
    throws java.rmi.RemoteException, javax.ejb.FinderException;
```

Note: Other methods for generating finder methods were available, but the one described here was the recommended method.

The FinderHelper interface has been replaced in WebSphere Studio Application Developer V5.1.1 with the use of the file **ibm-ejb-jar-ext.xmi**, located in the folder ejbModule/META-INF in the WebSphereCommerceServerExtensionsData project.

However, with WebSphere Studio Application Developer V5.1.1, you will not have to edit this file manually. To modify the finders for an entity bean, follow this procedure:

1. Switch to the **Assembly Descriptor** tab of the **EJB Deployment Descriptor**. (see , “Add the WCSSecurity role” on page 148 for details).
2. Select the entity bean in the list on the left that you wish to view or modify the finders for.
3. Scroll down to the **WebSphere Extensions** section.
4. The list box under the **Finders** headline contains the list of custom finders, defined for the selected EJB.

Note: All entity beans will have the finder findByPrimaryKey. This finder is not shown in this list.

Regenerating the deploy code

As you have made changes to your EJBs and the deployment descriptor, you must regenerate the deploy code:

1. Switch to the **J2EE Hierarchy** in the **J2EE Perspective**.
2. Expand **EJB Modules** and right-click **WebSphereCommerceServerExtensionsData: MyCustomEJBs**.
3. Select **Generate -> Deployment and RMIC Code...**
4. Ensure that the EJBs that you need to generate deployment code for are selected. Click **Select all** if you need to generate for all custom EJBs. Click **Finish**.

8.2.7 J2EE Connector Architecture

WebSphere Commerce Suite V5.1 used the Common Connector Framework, whereas WebSphere Commerce V5.6 uses the J2EE Connector Architecture.

If your WebSphere Commerce Suite V5.1 solution contains code that uses connectors, you must rewrite this code to use the J2EE Connector Architecture. For more information about the J2EE Connector Architecture, see:

<http://java.sun.com/j2ee/connector>

Note: This does not apply to code that uses the WebSphere Commerce Suite V5.1 messaging subsystem, as this subsystem abstracts the specifics about the implementation connector architecture.

8.2.8 Pricing

Some of the APIs for calculating and retrieving prices have changed from WebSphere Commerce Suite V5.1.

New task commands

Table 8-3 shows the three task commands, related to price calculations, that have changed since WebSphere Commerce Suite V5.1. The three task commands were introduced in WebSphere Commerce Suite V5.4.

Table 8-3 *Changed price task commands*

WebSphere Commerce Suite V5.1	WebSphere Commerce V5.6
GetBaseUnitPriceCmd	GetContractUnitPriceCmd
GetBaseSpecialPriceCmd	GetContractSpecialPriceCmd
GetProductBaseUnitPriceCmd	GetProductContractUnitPriceCmd

New data bean methods

In WebSphere Commerce V5.6, the following data beans were modified, replacing the method `getCalculatedPrice` with the method `getCalculatedContractPrice`:

- ▶ ItemDataBean
- ▶ PackageDataBean
- ▶ ProductDataBean
- ▶ CatalogEntryDataBean
- ▶ InterestItemDataBean
- ▶ BundleDataBean

The old data bean methods and task commands are still available in WebSphere Commerce V5.6, but we recommend that any code depending on these methods and commands be migrated to use the new methods and commands.

8.2.9 Product Advisor

WebSphere Commerce Suite V5.1 contains the following three files in the <wcs_home>\samples\web\pa directory:

- ▶ pc51.jsp
- ▶ pe51.jsp
- ▶ sa51.jsp

WebSphere Commerce V5.6, during the installation, provides newer versions for these sample files available in the following directories:

- ▶ For WebSphere Commerce V5.6 runtime:
 - <wc_home>\samples\web\pa
 - <was_home>installedApps\<nodename>\WC_<instancename>.ear\Stores.war
- ▶ For WebSphere Commerce V5.6 Toolkit:
 - <wctoolkit_home>\samples\web\pa
 - <wctoolkit_home>\workspace\Stores\Web Content

Where <wctoolkit_home> is the installation directory for WebSphere Commerce V5.6 Toolkit, <wc_home> is the installation directory for WebSphere Commerce V5.6, <was_home> is the installation directory for WebSphere Application Server V5.0.2, <nodename> is the WebSphere Application Server node name, and <instancename> is the name of the runtime instance.

If you based any development on these WebSphere Commerce Suite V5.1 sample JSP files, you must apply the changes in these JSPs to your customized JSPs.

Package name change

The package name for the following Product Advisor data type classes has been changed in WebSphere Commerce V5.6:

- ▶ DsCurrency
- ▶ DsDate
- ▶ DsDecimal
- ▶ DsDouble
- ▶ DsImage
- ▶ DsInteger
- ▶ DsString
- ▶ DsURLLink

In WebSphere Commerce Suite V5.1 these classes were in the package `com.ibm.commerce.datatype`, whereas in WebSphere Commerce V5.6 the package name has changed to `com.ibm.commerce.pa.datatype`.

Any code, referring to the types using the old package names must be updated to work with WebSphere Commerce V5.6.

8.2.10 Rule server administration commands

A number of changes have been made to the structure and behavior of the rule service administration commands. Refer to *Migration Guide for WebSphere Commerce Developer - migrating from WebSphere Commerce Studio V5.1* for details about the changes.

8.2.11 JSP and property file changes

Several improvements have been done to the login and register functionality. These improvements are reflected in new error codes which the JSPs must be able to deal with. For our example the new changes are split up in three sections:

1. Dealing with new error codes for logging in. This is dealt with in file **myAccount.jsp**
2. Dealing with new error codes when registering. This is dealt with in file **register.jsp**
3. Adding text for the new error codes for logging in and registering. This is dealt with in file **user_text_en_US.properties**

Note: The changes applied to the JSP files below are examples from our store which is based on the WebSphere Commerce Suite V5.1 sample store InFashion. For other stores the filenames may differ.

Refer to the WebSphere Commerce V5.6 InfoCenter under the relevant commands and their error codes to ensure that all necessary changes are applied.

Changes to myAccount.jsp

Several new error codes has been added to WebSphere Commerce V5.6 during login. To be able to deal with the new error codes, the file **myAccount.jsp** must be modified. In our example **myAccount.jsp** does not handle these new error codes and needs to be modified in order to display the error messages to the user during failed login.

To JSP file **myAccount.jsp** located in:

```
<was5_home>\installedApps\hostname\app_name\Stores.war\store_name
```

in our example:

D:\WebSphere\AppServer\installedApps\wcs56h\WC_demo.ear\Stores.war\MyStore

the following summary of changes has been added to improve the error handling:

- Add logic to check the error code returned and create error message to show user

Example 8-8 shows part of the content of the modified **myAccount.jsp** file. The changes are outlined in **bold**.

Example 8-8 The content of the modified myAccount.jsp file

```

        :
        :
        :
    }else if
    (strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_INVALID_PASSWORD) ==
    true) {
        strErrorMessage = usertext.getString("PASSWORD_INCORRECT");
    }else if
    (strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_DISABLED_ACCOUNT) ==
    true) {
        strErrorMessage = usertext.getString("ACCOUNT_LOCKED");
    } else if
    (strArrayAuth[0].equalsIgnoreCase(ECSecurityConstants.ERR_LOGON_NOT_ALLOWED) ==
    true) {
        strErrorMessage = usertext.getString("WAIT_TO_LOGIN");
    }
        :
        :
        :

```

Changes to Register.jsp

To improve the password security during account registration, several new checks has been implemented. Additionally for these checks, there are some new error codes. Our **Register.jsp** does not handle these new error codes and needs to be modified in order to display the error messages to the user during improper registering.

To JSP file **register.jsp** located in:

<WAS5_home>\installedApps\hostname\app_name\Stores.war\store_name

in our example:

D:\WebSphere\AppServer\installedApps\wcs56h\WC_demo.ear\Stores.war\MyStore

the following summary of changes has been added to improve the error handling:

- ▶ Add **import** statement to import the package which contains the new error codes
- ▶ Add logic to check the error code returned and create error message to show user

Example 8-9 shows part of the content of the modified **register.jsp** file. The changes are outlined in **bold**.

Example 8-9 The content of the modified register.jsp file

```

:
:
:
<%@ 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"%>
:
:
:
if (strErrorCode.equals(ECUserConstants.EC_ADDR_ERR_BAD_LASTNAME))
    strErrorMessage = usertext.getString("ERROR_MESSAGE45");
if
    (strErrorCode.equals(ECUserConstants.EC_UREG_ERR_MISSING_LOGONPASSWORDVERIFY))
        strErrorMessage = usertext.getString("ERROR_MESSAGE46");
if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMLLENGTH_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE21");
if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMDIGITS_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE22");
if (strErrorCode.equals(ECSecurityConstants.ERR_MINIMUMLETTERS_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE23");
if (strErrorCode.equals(ECSecurityConstants.ERR_USERIDMATCH_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE24");
if (strErrorCode.equals(ECSecurityConstants.ERR_REUSEOLD_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE25");
if (strErrorCode.equals(ECSecurityConstants.ERR_MAXCONSECUTIVECHAR_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE26");
if (strErrorCode.equals(ECSecurityConstants.ERR_MAXINTANCECHAR_PASSWORD))
    strErrorMessage = usertext.getString("PASS_ERROR_MESSAGE27");

//Redisplay what was entered when the
//invalid entry was submitted.
strLogonID = jhelper.getParameter(ECUserConstants.EC_UREG_LOGONID);
:
:

```

```

:
</body>

</html>

```

Changes to property files

The new error codes must have some descriptive text associated with them. This is done in the language-dependent properties file, located in the following directory:

```
<WAS5_home>\installedApps\hostname\app_name\Stores.war\WEB-INF\classes\store_name
```

In our example:

```
D:\WebSphere\AppServer\installedApps\wcs56h\WC_demo.ear\Stores.war\WEB-INF\classes\MyClasses
```

Example 8-10 shows the lines that must be added to the property file to support the changes made to **Register.jsp**, while Example 8-11 shows the lines that must be added to support the additions to **myAccount.jsp**.

Example 8-10 Error texts for file Register.jsp

```

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.

```

Example 8-11 Error texts for file myAccount.jsp

```
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.



Part 3

Production environment



Installing WebSphere Commerce V5.6

This chapter describes the procedure we used to install the WebSphere Commerce V5.6 runtime environment on a single-node.

The installation consists of the following part:

- ▶ Single-node runtime overview
- ▶ Installation
- ▶ Verifying the installation

9.1 Single-node runtime overview

The following sections describe the software and hardware we used for installing WebSphere Commerce V5.6.

9.1.1 Hardware

Ensure that your hardware meets the requirements described in 3.2.2, “Hardware and software prerequisites” on page 41.

We installed WebSphere Commerce V5.6 on a system with the following configuration:

- ▶ Intel Pentium 4 1.8 GHz
- ▶ 1.5 GB RAM
- ▶ 40GB hard drive
- ▶ 1024x768 Screen resolution
- ▶ CD-ROM drive

The time it took us to install and configure WebSphere Commerce V5.6 on this hardware is summarized in Table 9-1.

Table 9-1 Time used to install WebSphere Commerce V5.6 in our environment

Installation step	Time used (minutes)
WebSphere Commerce V5.6	80
DB2 Universal Database V8.1 fixpack 5	30
WebSphere Commerce instance creation	60
WebSphere Commerce Payments instance creation	30

9.1.2 Software

Ensure that your software meets the requirements described in 3.2.2, “Hardware and software prerequisites” on page 41.

Furthermore, you must read and follow the instructions regarding the preparation of the system and the DB2 Universal Database V8.1 user ID in *WebSphere Commerce V5.6 Installation Guide for Windows*.

Specifically, you must ensure that the user ID used while installing WebSphere Commerce V5.6 fulfills the following requirements:

- ▶ The user must be defined locally on the system and must belong to the local Administrators group.
- ▶ The user must have the following privileges assigned:
 - Act as part of the operating system
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token

Note: You must log off and log in again for these privileges to take effect.

- ▶ The user ID must not exceed 20 characters in length.
- ▶ The password must not exceed 14 characters in length.
- ▶ Both user ID and password may only contain characters from the set A-Z, a-z, 0-9.
- ▶ The user ID may not be any of the following: USERS, ADMINS, GUESTS, PUBLIC, or LOCAL, regardless of case.
- ▶ The user ID must not be the same as any Windows service name.

Important: If you choose to install the software from a directory on a local hard drive, or from a network location, you must ensure that the path to the WebSphere Application Server installation images not contain any long directory names (names that exceed the 8.3 naming scheme). An alternative solution is outlined in the following technote:

<http://www.ibm.com/support/docview.wss?uid=swg21157884>

Also, do not install from a UNC path, such as \\server\share. Map the share to a drive letter instead.

9.2 Installation

Follow the instructions in this section to install WebSphere Commerce V5.6 on a single-tier node, with DB2 Universal Database V8.1, WebSphere Application Server V5.0.2, and WebSphere Commerce Payments.

Important: After installing all the required software you must install WebSphere Commerce V5.6 Fixpack 1. It can be downloaded from the following URL:

<http://www-306.ibm.com/software/genservers/commerce/wcpe/support/>

9.2.1 Install WebSphere Commerce V5.6

Perform the following steps to install WebSphere Commerce V5.6:

1. Insert the WebSphere Commerce V5.6 CD 1 into your CD-ROM drive. If you have the autorun feature enabled, the launchpad will open. If the launchpad does not open, it can be started by running the **autorun.exe** file from the root of the installation CD, if desired.
2. After a while the language selection window for the launchpad will appear. Select your language and click **OK**. We selected **English**.
3. The WebSphere Commerce V5.6 LaunchPad will open. Click **Install Product**.

Tip: If you wish to start the Installation Wizard without starting the Launchpad, you can run the **setup.exe** file from the root of the installation CD 1.

4. After a while the language selection window for the main installation program will appear. Select your language and click **OK**. We selected **English**.

Note: The language selection window may open under the LaunchPad. Minimize the LaunchPad to see the language selection window.

5. The welcome window will appear. Click **Next**.
6. Select **I accept the terms in the license agreement** and click **Next**.
7. The Installation Wizard will display a warning window. Check to see if these warnings apply to you. Click **Next** when all problems have been resolved.
8. The system will prompt for the installation type. Select **Custom Installation** and click **Next**.
9. Select the components you want to install and click **Next**. Since we are installing a single-node system, we select the following components:
 - WebSphere Commerce Server
 - WebSphere Commerce Payments

- IBM DB2 Universal Database Enterprise Server Edition
- IBM HTTP Server

Note: The two remaining components, Remote WebSphere Commerce Configuration Manager client and WebSphere Application Server Web server plug-in, should not be selected as this is included in the already selected components.

10. The database and Web server selection window appears. Since we elected to install IBM DB2 and IBM HTTP Server in the previous window, it is not possible to modify the drop-down boxes. Click **Next**.
11. The destination paths window appears. Enter the desired destination paths for the components being installed and click **Next**.
We used the following values:
 - IBM DB2 Universal Database: D:\WebSphere\SQLLIB.
 - IBM HTTP Server: D:\WebSphere\HTTPServer.
 - IBM WebSphere Application Server: D:\WebSphere\AppServer.
 - IBM WebSphere Commerce Server: D:\WebSphere\CommerceServer.
12. You are prompted to enter the user ID and password for the administrative user. Enter the user ID and password for the current Windows user and click **Next**.
13. A confirmation message is shown, acknowledging that the user entered has the desired user privileges. Click **OK**.
14. The WebSphere Commerce Information Center language selection window appears. Select any additional languages that you may want and click **Next**.
15. The installation options confirmation window appears. Check the information and click **Next**.
16. Insert the IBM DB2 Universal Database V8.1 Fixpack 5 CD when the Installation Wizard prompts for it and click **Next**.
17. Insert the IBM WebSphere Application Server V5.0 CD when the Installation Wizard prompts for it and click **Next**.
18. Insert the IBM WebSphere Application Server Fixpack CD when the Installation Wizard prompts for it and click **Next**.

Note: During the installation of WebSphere Application Server V5.0.2 fixes, the installation progress bar may stay at 95% for a very long time. This is normal behavior. The progress of the fixpack installation can be followed by monitoring the contents of the <was_home>\logs\update directory and the file <was_home>\logs\WASFixPack.log, where <was_home> is the installation directory for WebSphere Application Server.

19. A window confirming that WebSphere Commerce V5.6 has been installed is shown. Click **Next**.
20. The Installation Wizard prompts for restarting the system. Ensure that **Restart now** is selected and click **Finish**.
21. After the system has restarted, the DB2 Universal Database V8.1 and WebSphere Commerce V5.6 First Steps wizards will appear. Close the DB2 Universal Database V8.1 First Steps wizard.
22. If the DB2 Universal Database V8.1 CD you supplied in step 16 on page 167 is not refreshed to fixpack 5 as mentioned previously, you must now apply DB2 Universal Database V8.1 fixpack 5.

9.2.2 Configuring IBM HTTP Server

In order to finish the installation, the IBM HTTP Server should be configured to use SSL and to improve performance, the Adaptive Fast Path Architecture (AFPA) Cache Accelerator should be enabled.

Enabling for SSL

In order for WebSphere Commerce V5.6 tools and for customer sites to work SSL has to be enabled. SSL certificates can either be self-signed or trusted (issued by a Certification Authority, CA). In our example we had a trusted SSL certificate, issued by VeriSign, which we copied from WebSphere Commerce Suite V5.1 and configured IBM HTTP Server to enable it. Please refer to WebSphere Commerce Developer Edition information center to see how SSL is enabled.

Enabling Cache accelerator

If you are using IBM HTTP Server v1.3.26.2 for WebSphere Commerce V5.6, you should enable the AFPA Cache Accelerator to improve performance for serving static content.

To enable AFPA, ensure that the following lines in the IBM HTTP Server <ihs_home>\conf\httpd.conf are activated, either by adding them or removing any line comment (#) in front of these lines:

```
Port 80
AfpaEnable
AfpaCache on
AfpaLogFile "/logs/afpalog" V-ECLF
```

If you enable AFPA, ensure that the following line is *not* active:

```
Listen 80
```

9.3 Verifying the installation

To verify the installation we will perform the following two steps:

- ▶ Verify installation log files
- ▶ Create test instance

9.3.1 Verify installation log files

Verify the installation log files. The details is outlined in the following sections:

- ▶ WebSphere Commerce installation logs
- ▶ DB2 Universal Database installation logs
- ▶ WebSphere Application Server installation logs
- ▶ IBM HTTP Server installation logs

Refer to the steps for verifying a custom installation in *WebSphere Commerce V5.6 Installation Guide for Windows* for details.

WebSphere Commerce installation logs

Examine the following log files to verify the base WebSphere Commerce installation:

```
<wc_home>\logs\install_<date>_<time>.log
<wc_home>\logs\wctrace_<date>_<time>.log
<wc_home>\logs\wcinstall.log
```

Where <wc_home> is the installation directory for WebSphere Commerce. If the installation fails, these files may be placed in the directory pointed to by the tmp Windows environment variable. The default value for this variable is the directory:

```
C:\Documents and Settings\<username>\Local Settings\Temp
```

Where <username> is the name of the currently logged on Windows user.

After a successful installation, the last message in the file **install_<date>_<time>.log** will contain the following text:

WebSphere Commerce installation complete.

DB2 Universal Database installation logs

Examine the following log files:

```
<wc_home>\logs\db2wi.log  
<wc_home>\logs\DB2.log
```

Where <wc_home> is the installation directory for WebSphere Commerce. If the installation fails, these files may be placed in the directory pointed to by the tmp Windows environment variable. The default value for this variable is the directory:

C:\Documents and Settings\<username>\Local Settings\Temp

Where <username> is the name of the currently logged on Windows user.

The last line of the **db2wi.log** file should be:

Product: DB2 Enterprise Server Edition -- Installation operation completed successfully.

Note: The following error messages may appear in the log file and can be safely ignored:

DEBUG: Error 2769: Custom Action StreamLibrarySCA did not close 1
MSIHANDLES.

1: Can not set the "SVCENAME=db2c_DB2" value in the Database Manager
Configuration File for the instance "DB2". Return code is "-104".

1: The Fast Connection Manager (FCM) base port was not specified for the
instance "DB2". Default parameters will be used.

1: The maximum number of logical nodes was not specified for the instance
"DB2". Default parameters will be used.

WebSphere Application Server installation logs

Examine the following log files to verify the WebSphere Application Server installation:

```
<was_home>\logs\http_plugin.log  
<was_home>\logs\log.txt  
<was_home>\logs\mq_install.log  
<was_home>\logs\WASFixeslog  
<was_home>\logs\WASFixes.err.log  
<was_home>\logs\WASFixPack.log
```

<was_home>\logs\WASFixPack.err.log

Where <was_home> is the installation directory for WebSphere Application Server. If the installation fails, these files may be placed in the directory pointed to by the tmp Windows environment variable. The default value for this variable is the directory:

C:\Documents and Settings\<username>\Local Settings\Temp

Where <username> is the name of the currently logged on Windows user.

The the **log.txt** file should contain the following line, confirming that the base WebSphere Application Server V5.0.2 installation has succeeded:

INSTFIN: The WebSphere 5.0 install is complete.

Examine the following file to confirm that the required fixes have been applied to WebSphere Application Server:

<was_home>\properties\version\BASE.product

The file should indicate the WebSphere Application Server version as 5.0.2.3.

Note: The following messages can safely be ignored, as they warn about situation that are not really error conditions:

- ▶ <was_home>\logs\log.txt:


```
(<date>,<time>), Setup.product.install,
com.installshield.product.service.product.PureJavaProductServiceImpl$I
nstallProduct, wrn, Did not replace installed object (IBM WebSphere
Application Server) with object (IBM WebSphere Application Server)
```
- ▶ <was_home>\logs\WASFixPack.log:


```
Fix pack: was502_cf3_win The specified fixpack contains updates for an
unavailable component (installation will continue):
prereq.db2
resources

Fix pack: was50_fp2_win Checking product features: IBM HTTP Server has
been verified successfully. The specified fixpack contains updates for
an unavailable component (installation will continue):
samp.messaging
```

Ensure that the following files were created, indicating that the related fixpack has been applied. All files are created in the directory <was_home>\properties\version\history:

- ▶ was_50_fp2_win.ptfApplied (only for Microsoft Windows 2000)

- ▶ was_50_fp2_win.ptfDriver (only for Microsoft Windows 2000)
- ▶ was_502_cf3_win.ptfApplied (only for Microsoft Windows 2000 and Microsoft Windows 2003 Server)
- ▶ was_502_cf3_win.ptfDriver (only for Microsoft Windows 2000 and Microsoft Windows 2003 Server)
- ▶ PQ83918.efixApplied
- ▶ PQ83918.efixDriver
- ▶ PQ84196.efixApplied
- ▶ PQ84196.efixDriver
- ▶ PQ85469.efixApplied
- ▶ PQ85469.efixDriver
- ▶ PQ85933.efixApplied
- ▶ PQ85933.efixDriver
- ▶ PQ86588.efixApplied
- ▶ PQ86588.efixDriver

IBM HTTP Server installation logs

Examine the following log files to verify the IBM HTTP Server installation:

```
<ihs_home>\logs\error.log  
<was_home>\logs\ihs_log.txt  
<was_home>\logs\WASFixes.log  
<was_home>\logs\WASFixPack.log
```

Where <ihs_home> is the installation directory for IBM HTTP Server and <was_home> is the installation directory for WebSphere Application Server. If the installation fails, these files may be placed in the directory pointed to by the tmp Windows environment variable. The default value for this variable is the directory:

```
C:\Documents and Settings\<username>\Local Settings\Temp
```

Where <username> is the name of the currently logged on Windows user.

Ensure that the following files are either empty or missing:

```
<was_home>\logs\WASFixes.err.log  
<was_home>\logs\WASFixPack.err.log
```

Note: The following log files are shared between WebSphere Application Server and IBM HTTP Server, and may thus contain information for both products:

```
<was_home>\logs\WASFixes.log  
<was_home>\logs\WASFixPack.log  
<was_home>\logs\WASFixes.err.log  
<was_home>\logs\WASFixPack.err.log
```

9.3.2 Create test instance

To verify that the installation is functional, it is recommended that you create a test instance for WebSphere Commerce and WebSphere Commerce Payments.

Create a WebSphere Commerce instance

1. Ensure that the Configuration Manager is started. Refer to Appendix A., “Managing WebSphere Commerce components” on page 239 for details.
2. Expand and right-click the tree **<WC_node> -> Commerce -> Instance List** and select **Create Instance...**
3. The **Instance** window of the Commerce Instance Creation Wizard opens. Enter the instance properties and click **Next**. We entered the following information:
 - Instance Name: test
 - Merchant Key: 0123456789abcdef
 - Site Admin ID: wcsadmin
 - Site Admin Password: www444w4
 - Confirm Site Admin Password: www444w4

Note: The chosen Site Administrator user ID and password is only for test purposes. It is strongly recommended to choose different values for a production site.

When choosing a password for the site administrator, the password must adhere to the following rules:

- ▶ Must contain at least one alphabetic character.
- ▶ Must contain at least one numeric character.
- ▶ The same character cannot occur more than four times in total.
- ▶ The same character cannot occur consecutively more than three times.

4. The database window appears. Ensure that the **Create a new DB2 database or Oracle tablespace** option is selected and that the **Remote** and **Advanced Options** check boxes are not checked. Enter the following values and click **Next**:

- Database administrator name: <userid>
- Database administrator password: <password>
- Confirm Database administrator password: <password>
- Database name: test
- Database type: DB2

Where <userid> and <password> are the user ID and password for the existing administrative user, which was used during the installation in 9.2, “Installation” on page 165.

5. The Schema window appears. Enter the same user ID and password used in the previous page and ensure that **Use staging server** is not checked and that **Set as active database** is checked. Click **Next**.

Note: It is recommended to use a different non-administrative user ID for a production system. For test purposes, we choose to reuse the administrative user ID for the schema owner.

6. It is not necessary to change the default values for the remaining pages. Keep clicking **Next** until the Auction page appears. Then click **Finish**.

The Commerce Instance Creation Wizard will now begin the instance creation. The progress can be monitored on the main Configuration Manager window which may be hidden behind the Commerce Instance Creation Wizard.

The Commerce Instance Creation Wizard may prompt you to restart the Web server as seen in Figure 9-1. Click **OK** to dismiss the dialog.

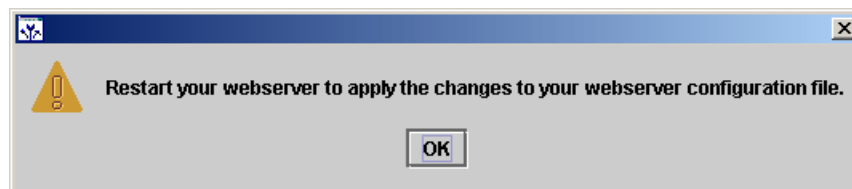


Figure 9-1 Web server restart prompt for commerce instance creation

If this happens, click **OK** to close the window and follow the directions in Appendix A, “Managing WebSphere Commerce components” on page 239 to restart the Web server.

When the instance has been successfully created, the following message should appear in the log panel of the Configuration Manager:

Event: Instance was successfully created.

Also, the message shown in Figure 9-2 will appear. Check that all messages in the dialog indicate success and click **OK**.

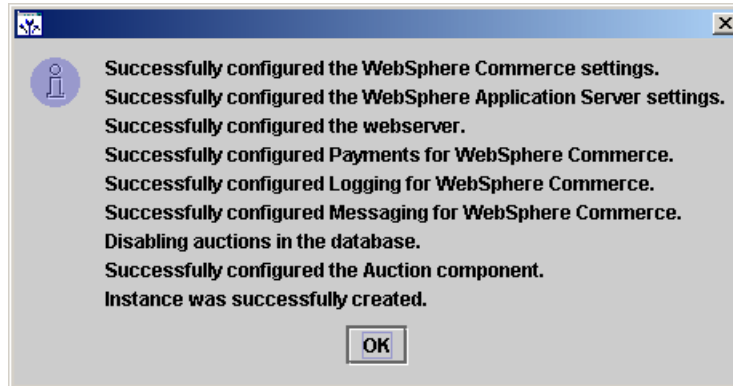


Figure 9-2 Commerce instance creation confirmation

Create a WebSphere Commerce Payments instance

1. Expand and right-click the tree **<WC_node> -> Payments -> Instance List** and select **Create Payments Instance**.
2. The **Instance** page of the Payments Instance Creation Wizard opens. Ensure that **Password Required for startup** is not checked and enter instance properties and click **Next**. We entered the following information:
 - Instance Name: testwpm
 - Instance Password: www44w4
 - Confirm Instance Password: www44w4
 - Site Admin ID: wcsadmin

Note: The chosen Instance Password is only for test purposes. It is strongly recommended to a different password for a production site.

When choosing an Instance Password, the password must adhere to the following rules:

- ▶ Must contain at least eight characters.
- ▶ Must contain at least one alphabetic character.
- ▶ Must contain at least one numeric character.
- ▶ The same character cannot occur more than four times in total.
- ▶ The same character cannot occur consecutively more than three times.

Likewise, it is also not recommended to uncheck the **Password Required for startup** check box on production systems.

3. The database page appears. Ensure that the **Create a new DB2 database or Oracle tablespace** option is selected and that the **Remote** check box is not checked. Enter the following values and click **Next**:

- Database administrator name: <userid>
- Database administrator password: <password>
- Confirm Database administrator password: <password>
- Database name: testwpm
- Database type: DB2

Where <userid> and <password> are the user ID and password for the existing administrative user, which was used during the installation in 9.2, “Installation” on page 165.

4. The Schema page appears. Enter the same user ID and password used in the previous page and ensure that **Use staging server** is not checked and that **Set as active database** is checked. Click **Next**.

Note: It is recommended to use a different non-administrative user ID for a production system. For test purposes, we choose to reuse the administrative user ID for the schema owner.

5. It is not necessary to change the default values for the two remaining pages. Click **Next** and **Finish**.

The Payments Instance Creation Wizard will now begin the instance creation. The progress can be monitored on the main Configuration Manager window which may be hidden behind the Payments Instance Creation Wizard.

The Payments Instance Creation Wizard may prompt you to restart the Web server as seen in Figure 9-3. Click **OK** to dismiss the dialog.

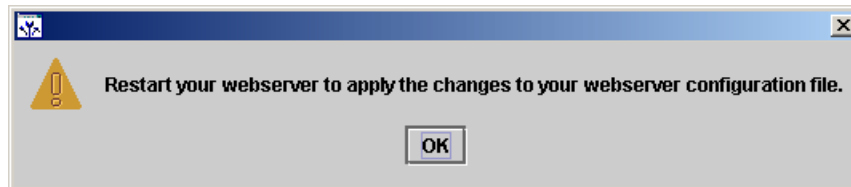


Figure 9-3 Web server restart prompt for payments instance creation

If this happens, click **OK** to close the window and follow the directions in Appendix A., "Managing WebSphere Commerce components" on page 239 to restart the Web server.

When the instance has been successfully created, the following message should appear in the log panel of the Configuration Manager:

Event: Instance was successfully created.

Also, the message shown in Figure 9-4 will appear. Check that all messages in the dialog indicate success and click **OK**.

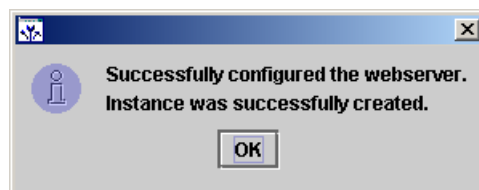


Figure 9-4 Payments instance creation confirmation

9.3.3 Removing test instances

After creating and verifying that the test instance is functional, the created instance is no longer needed and can be removed.

Removing WebSphere Commerce instance

The following steps describe how to remove a WebSphere Commerce instance.

1. Ensure that the instance is not running. Refer to Appendix A., “Managing WebSphere Commerce components” on page 239 for details about checking instance status and stopping WebSphere Commerce instances.
2. Remove the server from WebSphere Application Server by running the following command from the <wc_home>\bin directory:

```
rmCommerceServer <instancename>
```

Important: The directory, the script is executed from must be <wc_home>\bin. If executed from a different directory you will encounter the following error:

```
Exception in thread "main" java.lang.NoClassDefFoundError:  
com/ibm/commerce/config/components/WAS5AdminClient
```

3. Remove the instance database by running the following command from a DB2 command window:

```
db2 drop database <database_name>
```

Where <database_name> is the name of the instance database.

4. Remove the instance reference in the wcs_instances file:
 - a. Start the Configuration Manager as described in Appendix A., “Managing WebSphere Commerce components” on page 239.
 - b. Expand the tree until you find the instance to be deleted.
 - c. Right-click the instance name and select **Delete instance....**
 - d. You are asked if you really want to delete the instance. Click **Yes**.
 - e. A message informing that the instance was deleted appears. Click **OK**.
 - f. Close the Configuration Manager.

Tip: The Configuration Manager will remove the instance from the instance list, located in the file **wcs_instances**. Also, the file `<wc_home>\instances\<instancename>\xml\<instancename>.xml` will be renamed to `<instancename>.xml.bak`.

To accomplish the same result without opening the Configuration Manager, simply edit the file `<wc_home>\instances\wcs_instances`.

This file will contain a line like the following for each instance defined on the system:

```
<instancename>;local=<wc_home>\instances\<instancename>\xml\<instancename>.xml
```

Use a text editor to remove the line for the instance to be removed and save the file. The `<instancename>.xml` file will be removed in the following step.

5. Remove the instance directory. The default instance directory is:

```
<wc_home>\instances\<instancename>
```

If this has been changed from the default value, the new location is shown in the `wcs_instances` file as described in the previous step.

Removing WebSphere Commerce Payments instance

The following steps describe how to remove a WebSphere Commerce Payments instance.

1. Ensure that the instance is not running. Refer to Appendix A, “Managing WebSphere Commerce components” on page 239 for details about checking instance status and stopping WebSphere Commerce Payments instances.
2. Remove the server from WebSphere Application Server. Run the following command from the `<wc_home>\bin` directory:

```
rmPaymentsServer <instancename>
```

The **rmPaymentsServer.bat** script is shown in Example 9-1 and is also provided as part of the additional material for this Redbook. Refer to Appendix E, “Additional material” on page 273 for details.

Important: The `rmPaymentsServer.bat` script must be executed from the `<wc_home>\bin` directory. If executed from a different directory you will encounter the following error:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
com/ibm/commerce/config/components/WAS5AdminClient
```

Example 9-1 Sample rmPaymentServer.bat script

```

@echo off
REM Licensed Materials - Property of IBM
REM 5724-A18
REM (c) Copyright IBM Corp. 2000, 2004
REM US Government Users Restricted Rights - Use, duplication or
REM disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

setlocal

call config_env.bat
if exist %WAS_HOME%\bin\setupCmdLine.bat call %WAS_HOME%\bin\setupCmdLine.bat

set instanceName=%1
set serverName=%instanceName%_Commerce_Payments_Server

if "%instanceName%"==" " goto USAGE

set defs="-Dwas.install.root=%WAS_HOME%"
set defs=%defs% "-Dwas.repository.root=%CONFIG_ROOT%"
set defs=%defs% "-Dws.ext.dirs=%WAS_EXT_DIRS%"
set defs=%defs% -Dcom.ibm.CORBA.BootstrapHost=%COMPUTERNAME%
set defs=%defs% -Djava.security.policy="config.policy"

set classname=com.ibm.commerce.config.components.WAS5AdminClient
set params=-deleteInstance "%instanceName%"

"%JAVA_HOME%\bin\java" %PM_ARGS% %defs% %classname% %params%

"%WAS_HOME%\bin\WASService" -remove %serverName%
goto END

:USAGE
echo Usage: rmPaymentsServer (instanceName)
goto END

:END
endlocal

```

3. Remove the instance database. Issue the following command from a DB2 command window:

```
db2 drop database <database_name>
```

Where <database_name> is the name of the instance database.

4. Remove the instance reference in the wcs_instances file:
 - a. Start the Configuration Manager as described in Appendix A., "Managing WebSphere Commerce components" on page 239.

- b. Expand the tree until you find the instance to be deleted.
- c. Right-click the instance name and select **Delete instance....**
- d. You are asked if you really want to delete the instance. Click **Yes**.
- e. A message informing that the instance was deleted appears. Click **OK**.
- f. Close the Configuration Manager.

Tip: The Configuration Manager will remove the instance from the instance list, located in the file `wcs_instances`. Also, the file `<wc_home>\instances\<instancename>\xml\<instancename>.xml` will be renamed to `<instancename>.xml.bak` and the file `<wc_home>\payments\instances\<instancename>\PaymentsInstance.properties` will be deleted.

To accomplish the same result without opening the Configuration Manager, simply edit the file `<wc_home>\instances\wcs_instances`.

This file will contain a line like the following for each instance defined on the system:

```
<instancename>;local=<wc_home>\instances\<instancename>\xml\<instancename>.xml
```

Use a text editor to remove the line for the instance to be removed and save the file. The `<instancename>.xml` file will be removed in the following step.

5. Remove the instance directories. The default instance directories for a WebSphere Commerce Payments instance are:

```
<wc_home>\instances\<instancename>
```

and:

```
<wc_home>\payments\instances\<instancename>
```




Pre-migration steps

This chapter describes the actions needed to prepare the WebSphere Commerce Suite V5.1 elements prior to the migration.

The chapter is organized in the following sections:

- ▶ Preparation overview
- ▶ Instance preparation
- ▶ Database preparation

10.1 Preparation overview

This chapter describes the actions needed to prepare the WebSphere Commerce Suite V5.1 production environment prior to migrating to WebSphere Commerce V5.6.

It is recommended to prepare and migrate the elements in the following order:

1. Instance
2. Database

The migration steps are described in Chapter 11, “Migrating WebSphere Commerce components” on page 201

The following is a list of the specific sections that are covered. You can use this list as a checklist when you are performing the tasks in this chapter. Read and understand the chosen section completely, before you begin any of the tasks:

- ▶ Instance preparation
 - Updating the product information file
 - Prepare resources
 - Disable security on WebSphere Application Server
- ▶ Database preparation
 - Copying the database from Commerce V5.1 to Commerce V5.6
 - Updating database configuration values
 - Unsent messages
 - Orders status
 - Catalog
 - Members
 - Custom message types
 - Dropping foreign key references
 - Erroneous data in encrypted fields

In each section we analyze the element to see what needs to be changed in order to prepare it for the migration.

10.2 Instance preparation

This section describes the tasks needed to prepare our instance for the WebSphere Commerce V5.6 instance migration script. Some of the tasks needs to be performed in the WebSphere Commerce Suite V5.1 system and some in the WebSphere Commerce V5.6 system.

Note: The WebSphere Commerce Suite V5.1 environment does not need to be shut down in order to prepare the instance.

10.2.1 Updating the product information file

The product level information prior to WebSphere Commerce Suite V5.1.1.x is stored in SITE table. In WebSphere Commerce Suite V5.1.1.x and its successors, all product level information is stored in **product.xml** located in:

```
<wc51_home>\xml
```

The product file structure in WebSphere Commerce Suite V5.1 is not compatible with WebSphere Commerce V5.6 and cannot be migrated. To solve this incompatibility a template file is included in WebSphere Commerce V5.6 located in the following directory:

```
<wc56_home>\migration\product.xml.51.sample
```

To prepare the product information file for migration, complete the following steps:

1. On WebSphere Commerce Suite V5.1, rename the existing file (if any) **product.xml** to **product.xml.old**
2. Copy the **product.xml.51.sample** file from the WebSphere Commerce V5.6 system, in the following directory:

```
<wc56_home>\migration
```

To the following directory on the WebSphere Commerce Suite V5.1 system:

```
<wc51_home>\xml
```

3. Rename the copied file to **product.xml**.
4. Copy the **product.dtd** file on the WebSphere Commerce V5.6 system, from the following directory:

```
<wc56_home>\xml
```

To the following directory on the WebSphere Commerce Suite V5.1 system:

```
<wc51_home>\xml
```

5. Open the file **product.xml** in a text editor, for example, Notepad.

6. Update the relevant sections as shown in Example 10-1 on page 186. Ensure that you enter the appropriate levels for your scenario. Refer to Table 10-1, 10-2 and 10-3 for assistance on how to choose the values for your environment.
7. Update the path node with the path to your WebSphere Commerce Suite V5.1 installation, such that it corresponds to the WebSphere Commerce Suite V5.1 installation path specified in the installation path (*WCInstallDir*) specified in the *instance.xml*.

Note: No other tags need to be updated as they will be updated by the WCIM tool during instance migration.

8. Save and close the file.

Example 10-1 product.xml file content

```
<commerceserver>
  <name>IBM WebSphere Commerce Server</name>
  <edition>
    <name>edition_name</name>
  </edition>
  <version>5</version>
  <release>1</release>
  <modification>mod</modification>
  <fixpak>fixpak</fixpak>
  ...
  <install>
    ...
    <path>path</path>
    ...
  </install>
  ...
</commerceserver>
```

Table 10-1 Choosing the value for edition_name

WebSphere Commerce version	edition_name value
WebSphere Commerce Suite V5.1 Start Edition	Start
WebSphere Commerce Suite V5.1Pro Edition	Pro
WebSphere Commerce Suite V5.1 Business Edition	Business

Table 10-2 Choosing the value for mod

WebSphere Commerce version	mod value
WebSphere Commerce Suite V5.1.0.1	0
WebSphere Commerce Suite V5.1.1	1
WebSphere Commerce Suite V5.1.1.x	1

Table 10-3 Choosing the value for fixpack

WebSphere Commerce version	fixpack value
WebSphere Commerce Suite V5.1.0.1	1
WebSphere Commerce Suite V5.1.1.1	1
WebSphere Commerce Suite V5.1.1.2	2
WebSphere Commerce Suite V5.1.1.3	3
WebSphere Commerce Suite V5.1.1.4	4

In our example the **product.xml** file looked like this:

Example 10-2 Our product.xml file

```

<commerceserver>
  <name>IBM WebSphere Commerce Server</name>
  <edition>
    <name>Start</name>
  </edition>
  <version>5</version>
  <release>1</release>
  <modification>1</modification>
  <fixpak>2</fixpak>
  ...
<install>
  ...
  <path>d:\WebSphere\WCS</path>
  ...
</install>
  ...
</commerceserver>

```

10.2.2 Prepare resources

The WCIM tool will create a ZIP archive with the entire directory structure for the instance directory, that is all files under the directory

```
<wcs_home>\instances\<instancename>
```

Where <wcs_home> is the WebSphere Commerce Suite V5.1 installation directory and <instancename> is the name of your instance.

If you have WebSphere Commerce Suite V5.1 customized files and directories in other locations outside of the instance directory that you wish to have included in the ZIP archive, you must now copy these files and directories to subdirectories under the mentioned instance directory

In our example we did not have any customized files or directories.

Note: It is not a strict requirement that you copy customized files to the instance directory. The WCIM ZIP file is merely a convenient way to copy all necessary file assets from the WebSphere Commerce Suite V5.1 system to the WebSphere Commerce V5.6 system.

10.2.3 Disable security on WebSphere Application Server

If WebSphere Application Server V5.0.2 security is enabled on your WebSphere Commerce V5.6 system, you must disable it before migrating your instance by doing as follows:

1. Open the WebSphere Application Server Administration console for the WebSphere Commerce V5.6 system.
2. Click **Security > Global Security** and clear the **Enable Security** check box in the right panel
3. Click **OK** and save the changes
4. Restart the WebSphere Application Server administrative server on the WebSphere Commerce V5.6 system.

10.3 Database preparation

This section describes the tasks needed to prepare the database for the WebSphere Commerce V5.6 database migration script. Each task is split up in subsystems and can be applied in any order, only Section 10.3.1, “Unsent messages” on page 189 should be considered first.

We copied the database from the WebSphere Commerce Suite V5.1 system to the WebSphere Commerce V5.6 system and performed all the pre-migration preparation steps on it, recording all steps in a database script file.

Once our pre-migration script was built and tested on the database copy, we used the same script to perform pre-migration steps on the production database. This approach minimizes the downtime for the production system, as the production system will have to be down for the duration of the database migration.

Refer to Appendix C, “Migration scripts” on page 255 for the complete pre migration script containing all database changes shown in the following sections.

10.3.1 Unsent messages

In WebSphere Commerce Suite V5.1, when you send a SendTransacted message, the message is stored in the MSGSTORE table. Once the message has been sent, the entry is removed from the database. These entries cannot be reused in WebSphere Commerce V5.6 and the table must therefore be emptied prior to migration.

In order to clean all messages in the MSGSTORE table in the production database these steps are required:

1. Log on to the Administration Console as the Site Administrator.
2. At the Site or Store selection panel, select **Site**
3. Change the status of each transport to **Inactive** in the **Configuration - Transports** menu
4. Repeat the steps above to disable the transports for each store. Disabling the transport does not prevent messages in the MSGSTORE table from being sent. It only prevents new messages from being saved in the MSGSTORE table. The scheduler will attempt to deliver all messages in MSGSTORE. By default, the scheduler runs SendTransactedMsg job at 5 minute intervals, and the number of retries is 3. After 15 minutes, there should not be any message in the MSGSTORE table with retries that are greater than zero.

Note: Transports must be disabled for all the stores as well as the site

5. Check whether there are any remaining entries in MSGSTORE with retries greater than zero by connecting to the database and typing the following SQL statement:

```
select count(distinct msgid) from msgstore where retries > 0
```

The select statement shows how many messages WebSphere Commerce Suite V5.1 still needs to deliver. The result is one row, one column with a numeric result that is 0 (zero) or greater. If the result is 0, it means that there are no pending messages to be delivered and that the table is ready to be migrated. If the result is greater than 0, it means that WebSphere Commerce Suite V5.1 is still attempting to deliver remaining messages.

After this count reaches 0, it should not increment if you subsequently run this SQL. If you notice that the result keeps on incrementing, it means that not all transports have been shut down. Refer to the steps above to disable the transport. Once the SQL select statement returns 0, you can run the following:

```
select count(distinct msgid) from msgstore
where retries = 0 or retries = -1
```

Note: For the purpose of readers convenience, the above SQL statement is shown in two lines. When entering the command it must be entered on a single line.

This select statement determines whether there are any messages that have not been delivered. If the result is 0, there are no left over messages. If the result is anything larger than 0, there are messages left in the MSGSTORE table. It is recommended that you delete the remaining messages.

Note: On the replica database, clearing the MSGSTORE is simply done with an SQL command as shown in the following example:

```
delete from MSGSTORE
```

10.3.2 Copying the database from Commerce V5.1 to Commerce V5.6

To copy the database from one environment to the other you first need to make a backup of the database and then transfer the backup to the new environment. The following steps describe how to transfer the database from one environment to the other:

1. Shut down the WebSphere Application Server in order to close all its connections to the database.

To stop the WebSphere Application Server follow these steps:

- a. Open the WebSphere Application Server Administrative Console
- b. Expand the Administrative Domain
- c. Right click on the node and select **Stop**

- d. Click **Yes** on the Confirmation dialog window. The WebSphere Application Server Administrative Console will close automatically.

Ensure that all database background processes and connections are terminated by typing the following commands from a DB2 command line:

```
DB2 terminate
DB2 force application all
```

2. Take an offline backup of the database in the WebSphere Commerce Suite V5.1 environment by typing the following commands from a DB2 command line:

```
DB2 backup database <database name> to <target drive>
```

3. Start the WebSphere Application Server again.
4. Copy the backup folder to the WebSphere Commerce V5.6 system.
5. Create a database user in WebSphere Commerce V5.6 environment (in the operating system) with the same name, password and access privileges as the database user in WebSphere Commerce Suite V5.1.
6. Restore the database you just copied into the WebSphere Commerce V5.6 system by typing the following command from a DB2 command line:

```
DB2 restore database <source database name> from <backup directory>
taken at <backup timestamp> into <target database name>
```

Note: DB2 will automatically migrate the database from version 7 to version 8, after it has successfully restored the database.

The database can fail during migration (SQL2519N) if the database configuration values are too small. If the above error occurs, do the following to migrate the database manually:

1. Get the database configuration values


```
db2 get db cfg for <database>
```
2. Write down the values for **logfilsiz**, **logprimary** and **logsecond**
3. Increase the above value, for example by doubling the values:


```
db2 update db cfg for <WC_db> using logfilsiz <new value>
db2 update db cfg for <WC_db> using logprimary <new value>
db2 update db cfg for <WC_db> using logsecond <new value>
```
4. Migrate the database


```
db2 migrate database <WC_db> user <user> using <password>
```
5. Restore the configuration values to the original values.

10.3.3 Updating database configuration values

The database configuration values, such as logfile size, application heap and statement heap, must be updated up to WebSphere Commerce V5.6 level prior to running any migration scripts. A script file is provided with WebSphere Commerce V5.6 for updating the database configuration values:

```
<wc56_home>\bin\updateDB2Configuration.bat
```

The script is invoked as follows:

```
updateDB2Configuration <WC_db> [<logfile>]
```

Where <WC_db> is the name of the database to change configuration values for and <logfile> is an optional parameter, naming the file to use for logging the progress of the script.

In our example we called the as follows:

```
updateDB2Configuration mall c:\logs\updatedb2configuration.log
```

Once the script has executed, check the logs to ensure no errors has occurred. The script runs a series of SQL commands, so any SQL command failing is an indication of an error.

The script will update the database configuration values with the minimum recommended as shown in Table 10-4. Base on your database characteristics, your DBA should review and modify these values accordingly in the script before executing the script.

Table 10-4 DB2 configuration set by the updateDBConfiguration script

Parameter	Value
applheapsz	1000
stmtheap	60000
app_ctl_heap_sz	8192
locklist	2400
indexrec	RESTART
logfilsize	1000
logprimary	12
logsecond	10
pckcachesz	4096

Parameter	Value
catalogcache_sz	4096

Important: Ensure that all connections to the database are terminated before continuing. Otherwise the changes performed by the script in this section may not take effect.

10.3.4 Orders status

Orders still in state C (Complete - Payment has been authorized) will be reallocated from the inventory when the WebSphere Commerce V5.6 system is brought online.

To avoid this, you should change the status of the orders according to their true status.

The *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1* recommends that these orders be changed to status S (Shipped) to prevent order items from being allocated again in WebSphere Commerce V5.6.

The changes are done by executing the following SQL commands:

```
update orders set status='S' where status='C'
update orderitems set status='S' where status='C'
```

Note: This is not recommended if you are using WebSphere Commerce Payments, or a similar system to automatically capture payments for orders in state S, unless you can be sure that all these orders have actually been shipped.

You must determine the best approach for your scenario. Some of the orders in state C may actually be abandoned orders, and if this is the case, these should be changed to state X - cancelled.

10.3.5 Catalog

This sections describes the changes which needs to be applied for the catalog subsystem.

Master catalog

A master catalog is a central location to manage your store's merchandise. It is a single catalog containing everything your store has to offer. Master catalog was

introduced in WebSphere Commerce Suite V5.4. The master catalog rule is enforced and therefore will need to be added during the migration. The Product Management Tool only work with the master catalog.

The structural restrictions for a master catalog are:

- ▶ The master catalog must be a proper tree with no cycles
e.g The parent category A has a subcategory B. Subcategory B and any of B's subcategories must not have A as their child.
- ▶ A product cannot belong to more than one category
- ▶ All items under a product must belong to same category as the product belongs to

If the restrictions are not suitable for your catalog then sales catalog(s) can be created for navigational purposes. Sales catalogs have no structural restrictions.

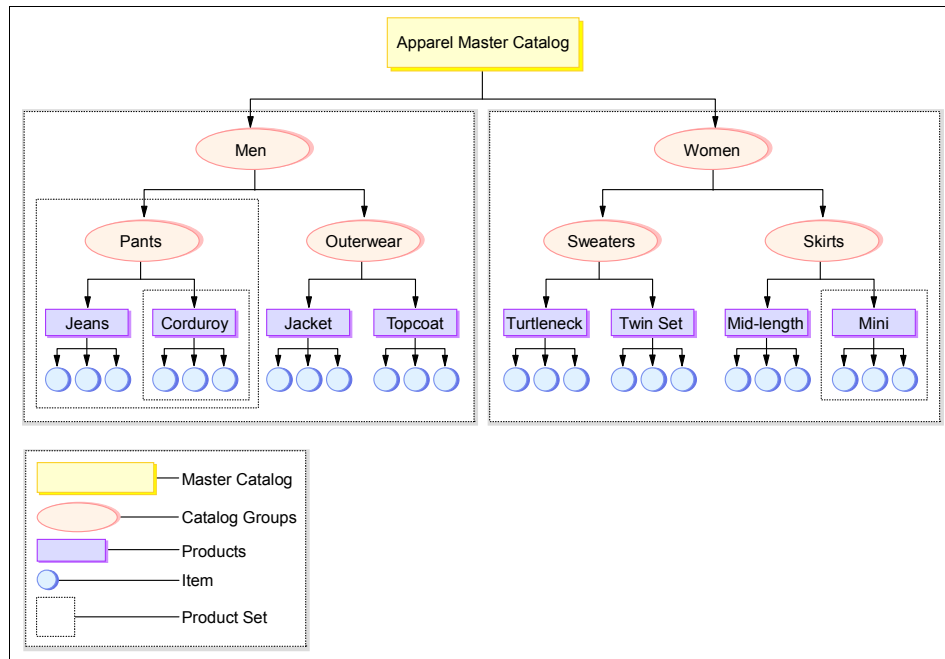


Figure 10-1 Example of a master catalog

If your store has more than one catalog you must choose one of the following:

- ▶ Set one of the existing catalogs as the master catalog, assuring that the restrictions mentioned above are applied.
- ▶ Create a new master catalog following the restrictions defined above.

In our example we only had one catalog which did not comply with the restrictions set for a master catalog as it had items, which belonged to more than one category. This meant that we had to create a new master catalog and use the existing catalog as a sales catalog.

To create a master catalog we performed these steps:

1. Create catalog

```
insert into catalog values (10002,-2000,'Master Catalog','Master Catalog')
```

2. Create catalog description

```
insert into catalogdsc values (10002,-1,'Master  
Catalog',NULL,NULL,NULL,NULL)
```

3. Create category (catalog group)

```
insert into catgroup values ((select max(catgroup_id) from  
catgroup)+1,-2000,'Master Category',0,NULL,NULL,NULL,NULL)
```

4. Create category description (catalog group description)

```
insert into catgrpdesc (catgroup_id,language_id, name, published,  
shortdescription) values ((select max(catgroup_id) from  
catgroup),-1,'Master Category',1,'Master Category description')
```

5. Create relation between catalog and category

```
insert into cattogrp values (10002,(select max(catgroup_id) from catgroup))
```

6. Create relation between store and catalog

```
insert into storecat values (10001,10001)  
insert into storecat values (10002,10001)
```

Note: The existing catalog (10001) was missing in STORECAT table, so we added it manually

7. Create relation between store and category (catalog group)

```
insert into storecgrp values (10001,(select max(catgroup_id) from  
catgroup))
```

8. Create relation between all catalog entries and category

```
insert into catgpenrel(catgroup_id, catalog_id, catentry_id, rule,  
sequence) select (select max(catgroup_id) from catgroup), 10002,  
catentry_id, '-', 0.0 from catentry
```

9. Update keys for catalog

```
update keys set counter=(select max(counter) from keys where  
tablename='catalog')+1 where tablename='catalog'
```

10. Update keys for category

```
update keys set counter=(select max(counter) from keys where  
tablename='catgroup')+1 where tablename='catgroup'
```

Product-item relationship

In WebSphere Commerce V5.6 all items must have a product. This rule is enforced as the Product Management Tool uses products as its starting point.

This rule can be applied in two ways (if needed):

1. Manually create products and associate items.

The benefits with this approach is less products and more organized product-item relations.

2. Let the migration script create one product (placeholder) for each item.

The benefits of this approach is that the script deals with creating the products, but the user may need to reorganize the products (placeholders) and their items after the migration.

In our example we let the migration script create the products (placeholders) for the items.

10.3.6 Members

This section describes the changes which needs to be applied for the member subsystem. This includes:

- ▶ Organizational structure
- ▶ Profile types
- ▶ Administrators
- ▶ Role changes

Organizational structure

Every user and organizational entity in WebSphere Commerce V5.6, with the exception of Root Organization, must have a parent organizational entity. This allows users and organizational entities to form a membership hierarchy.

The migration script will ensure that this is enforced, using the following rules:

- ▶ All organizations without a parent organization, that is organizations for which the MEMBER_ID column of ORGENTITY is set to null, will have Root Organization assigned as their parent organization.
- ▶ All users without an entry in the BUSPROF table will have Default Organization assigned as parent organization.

- ▶ Users with an entry in BUSPROF and with PROFILETYPE set to B (Business user) will be assigned the organization, pointed to by the ORGUNIT_ID or ORG_ID columns, as follows:
 - If ORGUNIT_ID is not null, the organization pointed to by ORGUNIT_ID, is used. as the user's parent organization
 - If ORGUNIT_ID is null and ORG_ID is not null, the organization pointed to by ORG_ID, is used as the user's parent organization.
 - If both ORGUNIT_ID and ORG_ID are null, the user is assigned Default Organization as parent organization.

Since business users should not be assigned Default Organization, we recommend that you ensure that all your business users an entry in BUSPROF with the right organization specified in ORGUNIT_ID.

Profile types

The WebSphere Commerce Suite V5.1 and WebSphere Commerce V5.6 schema allows for three profile types, as defined by the PROFILETYPE column of the USERS table:

- ▶ C - Consumer (B2C)
- ▶ B - Business (B2B)
- ▶ null - No profile data

It is recommended to clean up this column in a way that all users have either a Consumer or a Business profile. In other words, ensure that no rows in the USERS table have the PROFILETYPE column set to null.

In our scenario we changed the “null” profile type values to “B” by running the following commands:

1. Connect to the commerce database


```
DB2 connect to <WC_db> using <db_user_id> using <db_user_password>
```
2. Update the users profile type


```
DB2 update users set profiletype='B' where profiletype is null
```

Administrators

Administrators with register type S or A (Site administrator or Administrators, respectively) that also belongs to an access group must have their profile type set to B (Business user).

This is because all future administrators created in Organization Administration Console will have profile type B. This can be done with the following SQL:

```
DB2 update users set profiletype='B'
```

```
where registertype in ('A','S')
and users_id in (select users_id from accmbrgrp)
```

Role changes

Some of the roles in WebSphere Commerce Suite V5.1 are no longer available in WebSphere Commerce V5.6. Table 10-5 lists the WebSphere Commerce Suite V5.1 and the roles that should be used in WebSphere Commerce V5.6.

Table 10-5 Roles that are not included in WebSphere Commerce V5.6

WebSphere Commerce Suite V5.1 role	WebSphere Commerce V5.6 role
Order clerk (-5)	Customer Service Supervisor (-14)
Store Developer (-7)	Site Administrator (-1)
Customer role	Refer to WebSphere Commerce V5.6 migration guide for more information
Merchant	Seller
Merchandising Manager	Product Manager

Refer to *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1* for more information on the details regarding role migration.

The recommended approach to the roles, that are not included in WebSphere Commerce V5.6, is to replace them with roles which have the same function.

If a role is found by the migration script, and not found in WebSphere Commerce V5.6, it is migrated as a user-defined role.

Our application had two users with the following roles which needed to be changed:

1. Order Clerk
2. Merchandising Manager
3. Customer Service Representative

We found that Store Administrator would be a more suitable role for these two users. Therefore we assigned these two users the Store Administrator role instead.

1. Identify the users that should be assigned the Store Administrator role.

```
DB2 select * from accmbrgrp
```

In our example this command retrieved 6 records.

2. Delete all roles for user.

```
delete from accmbrgrp where users_id=<users_id>
```

3. Add Store Administrator as role for our users.

```
insert into accmbrgrp values (-6,<users_id>,-2000,NULL)
```

10.3.7 Custom message types

If you have created any custom message types in the MSGTYPES table, you may be required to update the MSGTYPE_ID column for your new types in order to avoid a conflict with new message types introduced in WebSphere Commerce V5.6. If you have not created any custom types, you can safely skip this sub-section.

As the MSGTYPE_ID column in the MSGTYPES must contain unique values, you must ensure that any custom message types that you have created do not have a MSGTYPE_ID the same as any new types in WebSphere Commerce V5.6. The simplest way to ensure that this does not happen is to update your custom types by adding 10000 to MSGTYPE_IDs. This must be done manually from a DB2 Command Window or the DB2 Command Center.

Failure to complete this step for custom message types can result in the database migration script failing to complete successfully.

10.3.8 Dropping foreign key references

If you have created any custom tables that have foreign key references to tables in the standard WebSphere Commerce data model, you must drop these foreign keys before proceeding with the migration. During the database migration, many tables are dropped, replaced with new versions and then re-populated with your data. If foreign key relationships exist, this may prevent the tables from being dropped, leading to the database migration script failing.

You must also drop any custom views that have been created in the database, as these may interfere with the operation of the database migration scripts.

Once the migration has completed successfully, you must recreate your foreign key relationships manually. If possible, while you are working on the migration, it is a good idea if a member of your team starts create a script that will recreate the references once the migration has completed. They will need to refer to the online help to check that the data model changes do not impact the referential integrity or appropriateness of the custom tables. The online help can be found at:

<http://publib.boulder.ibm.com/infocenter/wc56help/index.jsp>

10.3.9 Erroneous data in encrypted fields

Before migrating encrypted data, you should check the following fields in Table 10-6 for data that has a leading space and is unencrypted.

In WebSphere Commerce, in a field that may be encrypted, such as the field listed in Table 10-6, the presence of a leading space indicates that the field contains encrypted data.

If any of your unencrypted data has a leading space, due to a user entering this in the store for example, the migration tool will misinterpret this and try to decrypt the value, causing an error. The tables and fields to check are shown in Table 10-6. For further information about the behavior of the encrypted data migration script, please refer to Appendix C, "Migration scripts" on page 255.

Table 10-6 Tables and fields to check for erroneous leading spaces

Table	Columns to check	Notes
ORDPAYINFO	VALUE	Credit Card related information
ORDPAYMTHD	PAYDEVICE	Credit card related information



Migrating WebSphere Commerce components

This chapter describes the actions needed to migrate the WebSphere Commerce Suite V5.1 elements to the WebSphere Commerce V5.6 environment.

This chapter is organized in the following sections:

- ▶ Migration overview
- ▶ Migrating commerce instance configuration
- ▶ Migrating commerce database
- ▶ Migrating Payment Manager

11.1 Migration overview

This chapter describes the actions needed to migrate the WebSphere Commerce Suite V5.1 elements to the WebSphere Commerce V5.6 environment.

It is recommended to **prepare** and **migrate** the elements in the following order:

1. Instance
2. Database

The preparation steps are described in Chapter 10, “Pre-migration steps” on page 183.

The following is a list of the specific sections that are covered. You can use this list as a checklist when you are performing the tasks in this chapter. Read and understand the chosen section completely, before you begin any of the tasks:

- ▶ Migrating commerce instance configuration
- ▶ Migrating commerce database
 - Migrating the database
 - Choosing the master catalog
 - Migrating the encrypted data
- ▶ Migrating Payment Manager

In each section we describe the steps needed to migrate the individual element.

11.2 Migrating commerce instance configuration

Before the migration can take place ensure that the steps outlined in Section 10.2, “Instance preparation” on page 185 has been completed.

To migrate an instance the WCIM tool is provided with WebSphere Commerce V5.6.

It is recommended to understand the behavior of the WCIM tool before continuing on migrating the instance . The behavior of the WCIM tool is described in Appendix C, “Migration scripts” on page 255

The instance can be migrated in one of the following ways:

- ▶ Switch-Over
- ▶ In-place

Note: Although Co-existence is supported by the WCIM tool, it is not supported in Microsoft Windows platform.

For more information on instance migration approaches please refer to Chapter 3, “Migration Strategy and Planning” on page 37

We choose to do an Switch-Over migration as we had available hardware running WebSphere Commerce V5.6.

The following are the high level steps that needs to be performed during a Switch-Over migration:

1. On your WebSphere Commerce V5.6 system, create a package that contain all files needed to execute the WCIM tool on WebSphere Commerce Suite V5.1 system.
2. Transfer the package from your WebSphere Commerce V5.6 system to your WebSphere Commerce Suite V5.1 system.
3. Extract the package on WebSphere Commerce Suite V5.1 system.
4. On your WebSphere Commerce Suite V5.1 system, backup your current WebSphere Commerce instance and the instance-related assets using WCIM.
5. Transfer the instance backup produced by WCIM from WebSphere Commerce Suite V5.1 system to your target WebSphere Commerce V5.6 system.
6. Migrate instance using WCIM.
7. Verify that the instance was migrated successfully.

The steps must be repeated on each instance, if more than one instance is to be migrated.

During the steps performed on WebSphere Commerce Suite V5.1 the server does not need to be shut down, but it is recommended to shut it down or do it at a time when there is minimal load on the server.

The migration script WCIM, does not need the WebSphere Commerce V5.6 server running.

11.2.1 Migrating the instance

This section describe the tasks needed to migrate an WebSphere Commerce instance.

The WCIM tool **wcim.bat**, is provided to do the instance migration. WCIM is located in the following directory:

```
<wc56_home>\bin\
```

In our example:

```
D:\WebSphere\CommerceServer56\bin
```

Note: The WCIM tool must be run from the WebSphere Commerce V5.6 bin subdirectory.

Refer to *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1* for a complete syntax information.

To migrate the instance to WebSphere Commerce V5.6 complete the following steps:

1. Create a WCIM package file

Run the WCIM tool on the WebSphere Commerce V5.6 system with the following parameters:

```
wcim.bat -component wcim
```

from the following directory:

```
<wc56_home>\bin\
```

This will generate the following file:

```
<wc56_home>\temp\zip\backupwcim.zip
```

2. Transfer the package.

Copy the **backupwcim.zip** from you the WebSphere Commerce V5.6 system to the working directory on WebSphere Commerce Suite V5.1 system.

```
<wc51_home>\temp
```

If this temp folder does not exist, it must be created manually.

3. Unpack **backupwcim.zip**

Unpack the contents of the **backupwcim.zip** file into the working directory on WebSphere Commerce Suite V5.1 system.

```
<wc51_home>\temp
```

4. Backup instance

Before running this step ensure that you have the following information available.

Note: all the information required for this step is derived from the WebSphere Commerce Suite V5.1 system.

- The WebSphere Application Server home directory <was35_home>.
- The WebSphere Application Server administration node name.
- The WebSphere Application Server node name.
- Whether the IBM HTTP Server is installed locally or not.
- The IBM HTTP Server home directory <ihs_home>.

On the WebSphere Commerce Suite V5.1 system, run the WCIM tool with the following parameters to backup your existing instance:

```
wcim.bat -component wc -from 51 -backup remote -instance <instance_name>
```

from the following directory:

```
<wc51_home>\temp\bin\
```

In our example:

```
wcim.bat -component wc -from 51 -backup remote -instance demo
```

- Enter the previous WebSphere Application Server home directory:

```
D:\WebSphere\AppServer
```

- Enter the previous WebSphere Application Server administration node name:

```
wcs51
```

- Enter the previous WebSphere Application Server node name:

```
wcs51
```

- Is a previous IBM HTTP Server installed locally? [DEFAULT yes]

```
yes
```

- Enter the previous IBM HTTP Server home directory:

```
D:\WebSphere\HTTP
```

Running the above command will produce the following file:

```
<wc51_home>\temp\instance_backup.zip
```

The **instance_backup.zip** file contains a single file, **backupwc_51_<instance name>.zip**, which is used to deploy the instance on WebSphere Commerce V5.6.

5. Transfer and unpack the backup file

Copy the generated file, **instance_backup.zip**, to the working directory on the WebSphere Commerce V5.6:

```
<wc56_home>\temp
```

Unpack the **instance_backup.zip** file in the working directory of WebSphere Commerce V5.6 machine. This will produce the following file:

```
<wc56_home>\temp\zip\backupwc_51_instance_name.zip
```

6. Migrate the instance

Before running this step ensure that you have the following information available:

- Whether the IBM HTTP Server is installed locally or not.
- The IBM HTTP Server home directory <ihs_home>, if the IBM HTTP server is installed locally.
- A new Web server host name. If the Web server is installed locally on the WebSphere Commerce V5.6 system, the default host name can be accepted.

Note: The Web server host name used by WebSphere Commerce Suite V5.1 cannot be used.

- Whether the WebSphere Commerce Suite V5.1 instance and the new WebSphere Commerce V5.6 instance are using the same database server.
- A new database name for DB2 databases or a new service ID for Oracle databases if the same database server is to be used for WebSphere Commerce Suite V5.1 and WebSphere Commerce V5.6.

Note: all the information required for this step is derived from the WebSphere Commerce V5.6 system.

On the WebSphere Commerce V5.6 system, run the WCIM tool from the following directory:

```
<wc56_home>\bin\
```

with the following parameters to migrate your existing instance :

```
wcim.bat -component wc -from 51 -migration remote -instance <instance name>
```

In our example:

```
wcim.bat -component wc -from 51 -migration remote -instance demo
```

- Is a new IBM HTTP Server installed locally? [DEFAULT yes]
yes
- Enter the new IBM HTTP Server home directory:
D:\WebSphere\IBMHTTPServer
- Enter a new Web server host name: [DEFAULT wc56.itso.ral.ibm.com]
wc56.itso.ral.ibm.com
- Are the old instance and the new instance using the same database server? [DEFAULT yes]
no

Running the above command will migrate your instance to the WebSphere Commerce V5.6 level.

7. Verify the migration

A number of log files are generated by WCIM script. They are located in:

```
<wc56_home>\temp\logs\
```

The following log files generated by the script, must be checked to ensure a successful instance migration. The log files are:

- wcim.migration.wc.<instance name>.51.timestamp.log

Ensure that the following line is at the end of the file:

```
Event: WCIM has completed the job(s) assigned successfully
```

- instanceXmlMigration.log

Ensure that the following lines is at the end of the file:

```
Info: WebSphere Commerce instance configuration migrated  
successfully.
```

```
.  
.
.
```

```
Info: WebServer configuration migrated successfully.
```

11.3 Migrating commerce database

We created our own migration scripts to ease the database migration, these custom scripts assist you in the pre-migration, migration and post-migration steps. Once our migration script was built and tested on a test environment, we used the same script to perform the migration on the production database. This

approach would result in minimum hassle. All the database migration tasks were done in WebSphere Commerce V5.6 system. Refer to Appendix C, “Migration scripts” on page 255 for details on custom scripts behavior.

Here is the high level steps and order required to migrate the WebSphere Commerce database:

1. Migrating the database
2. Choosing the master catalog
3. Migrating the encrypted data

Note: The custom migration scripts mentioned in this book are not necessary to accomplish the WebSphere Commerce migration, we created those scripts with the only purpose of automating mainly the pre and post migration tasks. These scripts become more useful if you have to repeat these steps several times when you have several environments to migrate.

11.3.1 Migrating the database

Before the migration can take place ensure that the steps outlined in Section 10.3, “Database preparation” on page 188 has been completed.

It is recommended to understand the behavior of the database migration tool before proceeding with the database migration. The behavior of the database migration tool is described in Appendix C, “Migration scripts” on page 255.

The migration script can check if everything is correct before it makes any changes. This can be done with **precheck** option set. This executes the WebSphere Commerce V5.6 migration script in “read-only” mode.

The migration script **migratedb.bat** for the database is located in the following directory:

```
<wc56_home>\bin\
```

In our example:

```
D:\WebSphere\CommerceServer56\bin
```

The script receives the following parameters:

```
-dbtype <database_type>  
-dbname <database_name>  
-dbuser <user>  
-dbpass <password>  
-from <version>  
-instance <instance_name>
```

-precheck

Table 11-1 details the mandatory parameters and values for the database migration script.

Note: For **migratedb.bat** script to succeed the following rules must be applied:

1. **migratedb.bat** must be executed from a DB2 command window (db2cmd)
2. All the connections to the database must be terminated prior to and during the script execution. Prior to running this script all connections can be closed by issuing the following command:

```
db2 force application all
```

Table 11-1 Mandatory migration script parameters

Parameter name	Parameter value
dbtype	The database type: 'db2' for DB2 'oracle' for Oracle 'os400' for DB2/400 database on the iSeries platform 'db2j' for Cloudscape database 'os390' for DB2 for z/OS database.
dbname	The database name
dbuser	The database user
dbpass	The password for the database user
from	The version number migrating from: '51' for WebSphere Commerce Suite V5.1 database
instance	The WebSphere Commerce Suite V5.1 instance name

In our example:

```
migratedb -dbtype db2 -dbname mall -dbuser dbusr01 -dbpass dbusr01pwd -from  
51 -instance demo -precheck
```

The script will run and report any errors found and it will not make any changes to the database. Correction of the errors (depending on the code number) will either be mandatory or optional. The codes and their explanation can be found in the WebSphere Commerce V5.6 migration guides.

The log file generated by the migration script is found here:

```
<wc56_home>\logs\migration\instancename\migrateddb51_timestamp.log
```

In our example:

D:\WebSphere\CommerceServer56\logs\migration*instancename*\migrateddb51_*time stamp*.log

Example 11-1 A Sample of the migration log file with the precheck option set

```
[2004.04.30 11:38:29] Info : Logging started in VERBOSE mode.
[2004.04.30 11:38:29] Info : DTD path: ...
[2004.04.30 11:38:29] Event: Loading plan:
D:\WEBSPH~1\COMMER~1\schema\migration\config\DataMigrationPlan51.xml
[2004.04.30 11:38:32] Info : Plan loaded.
[2004.04.30 11:38:32] Info : dbname: mall
[2004.04.30 11:38:32] Info : dbuser: dbusr01
[2004.04.30 11:38:32] Info : dbtype: db2
[2004.04.30 11:38:32] Info : instance name: demo
[2004.04.30 11:38:32] Info : schema name: dbusr01
[2004.04.30 11:43:08] Info : set schema dbusr01
[2004.04.30 11:43:14] Info : SELECT PRODUCTVERSION FROM SITE
[2004.04.30 11:43:16] Info : SELECT LANGUAGE_ID FROM LANGUAGE WHERE LOCALENAME='en_US'
[2004.04.30 11:43:17] Info : Init Migrator ends.
[2004.04.30 11:43:17] Event: Pre migration begins...
[2004.04.30 11:43:21] Event: Executing command: precheck
[2004.04.30 11:43:21] Event: running java program:
com.ibm.commerce.migration.command.PreCheckCommand
[2004.04.30 11:43:23] Info : select count(*) from bzrpentstg where LENGTH(value) > 4000
[2004.04.30 11:43:24] Info : select count(*) from calcodedsc where LENGTH(longdescription) >
4000
.
.
.
[2004.04.30 11:43:14] Warning: In order to migrate one of the biggest tables "ORDERITEMS" in
the database, the database migration script needs at least 303 MB in the database log space. It
also needs at least 303 MB free disk space in the tablespace. Otherwise, the database migration
may fail.
.
.
.
[2004.04.30 11:43:35] Warning: [401] You have some items without parent products. You may
create a parent product for each item if you do not want default migration behaviour.
.
.
.
[2004.04.30 11:43:35] Info : select count(*) from storecgry where name is null
[2004.04.30 11:43:35] Event: In the database prechecking, there are some warnings. See log file
for details.
[2004.04.30 11:43:35] Info : Executed command: precheck
[2004.04.30 11:43:35] Event: Pre migration ends.
[2004.04.30 11:43:35] Event: Migration has terminated successfully.
```

In the above example the migration script has generated two warning (outlined in bold). The first one is to warn you that the transaction log size would get full and cause the migration script to fail. To solve this we updated the size of the transaction log to 30,000 blocks which should be enough for even very large databases.

```
db2 update db configuration for <database name> using logfilsiz 30000
```

The second warning is originated because of the missing products for the items (explained in detail in Product-item relationship in Section 10.3.5, “Catalog” on page 193). We choose to ignore this warning as the migration script will create the products for us.

Once all the errors has been corrected, the script can be run again with the same syntax as before, but with the **precheck** option removed. The log file will be the same:

```
<wc56_home>\logs\migration\instancename\migrateddb51_timestamp.log
```

Example 11-2 A small extract of the migration log file

```
[2004.04.29 20:48:22] Info : Logging started in VERBOSE mode.
[2004.04.29 20:48:22] Info : DTD path: ...
[2004.04.29 20:48:22] Event: Loading plan:
D:\WEBSPH~1\COMMER~1\schema\migration\config\DataMigrationPlan51.xml
[2004.04.29 20:48:25] Info : Plan loaded.
[2004.04.29 20:48:25] Info : dbname: mall
[2004.04.29 20:48:25] Info : dbuser: dbusr01
[2004.04.29 20:48:25] Info : dbtype: db2
[2004.04.29 20:48:25] Info : instance name: demo
[2004.04.29 20:48:25] Info : schema name: dbusr01
[2004.04.29 20:52:59] Info : set schema dbusr01
[2004.04.29 20:53:03] Info : SELECT PRODUCTVERSION FROM SITE
[2004.04.29 20:53:08] Info : SELECT LANGUAGE_ID FROM LANGUAGE WHERE LOCALENAME='en_US'
[2004.04.29 20:53:08] Info : Init Migrator ends.
[2004.04.29 20:53:09] Event: Pre migration begins...
[2004.04.29 20:53:13] Event: Executing command: precheck
[2004.04.29 20:53:13] Event: running java program:
com.ibm.commerce.migration.command.PreCheckCommand
.
.
.
[2004.04.29 20:53:31] Warning: [401] You have some items without parent products. You may
create a parent product for each item if you do not want default migration behaviour.
.
.
.
[2004.04.29 21:08:36] Info : CREATE INDEX I0000433 ON accmbrgrp ( owner_id ASC )
```

```

[2004.04.29 21:08:36] Error: [IBM][CLI Driver][DB2/NT] SQL0605W The index was not created
because the index "DBUSR01.I0000433" already exists with the required description.
SQLSTATE=01550
.
.
.
[2004.04.29 21:09:35] Info : CREATE INDEX I0000539 ON contract ( member_id ASC )
[2004.04.29 21:09:35] Error: [IBM][CLI Driver][DB2/NT] SQL0601N The name of the object to be
created is identical to the existing name "DBUSR01.I0000539" of type "INDEX". SQLSTATE=42710
.
.
.
[2004.04.29 21:53:37] Info : SELECT MAX(trdrefamt_id) as maxId FROM trdrefamt
[2004.04.29 21:53:37] Info : SELECT * FROM keys WHERE KEYS_ID=-1214
[2004.04.29 21:53:37] Info : Executed command: update.wcs.bootstrap.keys
[2004.04.29 21:53:37] Event: Post migration ends.
[2004.04.29 21:53:37] Event: Migration has terminated successfully.

```

When we checked the log file after running the migration script we found numerous errors split up in 2 SQL codes '**SQL0605W**' and '**SQL0601N**'.

Error SQL0605W

A sample of this error in the log:

```

[2004.04.29 21:08:36] Info : CREATE INDEX I0000433 ON accmbrgrp ( owner_id
ASC )
[2004.04.29 21:08:36] Error: [IBM][CLI Driver][DB2/NT] SQL0605W The index
was not created because the index "DBUSR01.I0000433" already exists with
the required description. SQLSTATE=01550

```

This SQL code can be ignored as its more a info message than error. The description of the SQL code clearly tells us that the index already exists with the required criteria and thus can be ignored.

Error SQL0601N

A sample of this error in the log:

```

[2004.04.29 21:09:35] Info : CREATE INDEX I0000539 ON contract ( member_id
ASC )
[2004.04.29 21:09:35] Error: [IBM][CLI Driver][DB2/NT] SQL0601N The name
of the object to be created is identical to the existing name
"DBUSR01.I0000539" of type "INDEX". SQLSTATE=42710

```

This error can have one of the following meanings:

- ▶ The index is missing (regardless of the message saying that it already exists). To correct this the index has to be created manually:

```
db2 create index <index name> on <table name> ( <column names> <order>)
```

- The index exists but on the wrong table. Typically it would exist on the same tablename with an '_OLD' appended to it (e.g for above mentioned table the index would exist on 'CONTRACT_OLD' instead of on 'CONTRACT'). To correct this error the index must be dropped and recreated on the right table:

```
db2 drop index <index name>
```

```
db2 create index <index name> on <table name> ( <column names> <order>)
```

These are the SQL commands we used in our example to correct the errors after running the migration script:

```
db2 create index I0000511 on campaign ( storeent_id asc)
```

```
db2 drop index I0000539
```

```
db2 create index I0000539 on contract ( member_id asc)
```

```
db2 create index I0000588 on initiative ( storeent_id asc)
```

```
db2 drop index I0000824
```

```
db2 create index I0000824 on userpvcdev ( users_id asc)
```

Note: review the log file for the values for the following parameters <index name>, <table name>, <column names> and <order>.

11.3.2 Choosing the master catalog

When the migrate database script is running it will generate some SQL scripts which can be used after migration. These scripts are located in the following directory:

```
<wc56_home>\instances\demo\migration
```

One of the SQL scripts generated, is the **choosemc.sql**. This SQL is used for selecting one of the catalogs as the master catalog. This script is only generated if the migrate database scripts detects more than one catalog in the WebSphere Commerce system. If only one catalog is found, it is set as the master catalog. Prior to running the script it must be modified to the correct catalog IDs. This is done by replacing every instance of MASTERCATALOG_ID with the catalog ID which is the master catalog. In our example the content of the file was updated as described in the following example.

Example 11-3 The content of the modified choosemc.sql file

```
--store :10001 has 2 catalogs.
--catalog:10001
--catalog:10002
```

```
--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=10002 and  
storeent_id=10001;  
insert into catgrptpc (catgroup_id,catalog_id,tradeposcn_id) values  
(0,10002,10001);
```

To execute the file, run the following from a DB2 command window:

```
db2 -tvf choosemc.sql
```

Ensure that the script has run successfully by examining the output.

11.3.3 Migrating the encrypted data

Once the database is migrated there are some additional steps to complete the migration.

Merchant key and encrypted data

The merchant key in WebSphere Commerce Suite V5.1 can be either default or custom. A default merchant key is a fixed value selected by WebSphere Commerce Suite V5.1 while the custom key is defined by the user. A default merchant key is not valid in WebSphere Commerce V5.6 and must be changed to a user defined merchant key. Since the encryption behavior has changed for sensitive data (as described below) the encrypted information must be re-encrypted using the new behavior to ensure that encrypted data is valid.

Users

Users stored in USERREG table (registered users) have a logon ID and a password. In WebSphere Commerce Suite V5.1 the password is encrypted using the merchant key and can be decrypted using the merchant key. In WebSphere Commerce V5.6 the USERREG has an extra column SALT (introduced in WebSphere Commerce Suite V5.4), which is used as an extra factor in the encryption of the password. The SALT key is added to the password and a one way hash key (SHA-1) is returned from the concatted string. This hash key is then encrypted. This method ensures that it is almost impossible to decrypt and retain the password. Because of this extra security the passwords must be decrypted and re-encrypted with a random SALT key and using a one way hash key.

Credit cards

Credit card data stored in ORDPAYINFO, ORDPAYMTHD and PATTRVALUE contains sensitive data and will typically be encrypted. The variable *PDIEncrypt* in the instance file indicates if credit card data is encrypted. Prior to WebSphere Commerce Suite V5.1.1.2 the merchant key was not used to encrypt credit card

data, instead a system key was used. Since this behavior has changed the data must be re-encrypted.

To summarize, the encrypted data must be re-encrypted because:

- ▶ Salt key is introduced for passwords
- ▶ One way hash keys are used for passwords
- ▶ Prior to WebSphere Commerce Suite V5.1.1.2 a system key was used for encrypting credit card data instead of using the merchant key
- ▶ A default merchant key is not valid and a new merchant key must be defined and therefore the encrypted data must be re-encrypted with the new merchant key

Encrypted data is migrated by calling the **migrateencryptedinfo.bat** script located in:

```
<wc56_home>\bin\
```

In our example:

```
D:\WebSphere\CommerceServer56\bin
```

The script must have the following mandatory parameters:

```
-dbtype
```

The optional parameters are:

```
-instance  
-current_key  
-new_key
```

Table 11-2 Encrypted data migration script parameters

Parameter name	Parameter value
dbtype	The database type. 'db2' for DB2 'oracle' for Oracle
instance	The WebSphere Commerce Suite V5.1 instance name. This parameter is optional as long as there is only one instance existing.
current_key	The current merchant key
new_key	The new merchant key (must conform with WebSphere Commerce V5.6 merchant key restrictions)

The script behaves in different ways depending on what parameters it receives.

Table 11-3 Encrypted data migration script parameters and behavior

Behavior needed	Parameters to include
A non-default WebSphere Commerce Suite V5.1 merchant key is used and should be reused.	dbtype instance (optional) example: migrateencryptedinfo db2 demo
A default WebSphere Commerce Suite V5.1 merchant key is used and needs to be redefined as a custom merchant key. Note: The current default merchant key is not used as a parameter	dbtype instance new_key example: migrateencryptedinfo db2 demo abcdef0123456789
A non-default WebSphere Commerce Suite V5.1 merchant key is used but the user wants to define a new merchant key.	dbtype instance current_key new_key example: migrateencryptedinfo db2 demo 0123456789abcdef abcdef0123456789

In our example we wanted to reencrypt the data with the current merchant key, so we ran the script with the following parameters:

```
migrateEncryptedInfo db2 demo
```

Once the migration script is finished the following logs will be created:

- ▶ CCInfoMigration.log
- ▶ MKChangeUserAndCCInfoMigration.log
- ▶ MigrateEncryptedInfoError.log

in directory:

```
<wc56_home>\logs
```

Note: At the time of writing this book this script had an error when managing the logs. The log files after execution are empty, but still can be monitored during the execution time. We used the **tail** command (unix command) as it would show the content on the fly (to have this unix command in a windows environment, an additional unix tool package for windows needs to be installed).

Review the logs to ensure that no errors occurred while migrating the encrypted data. The following errors are typical caused by an incorrect merchant key:

```
%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) snot  
multiple of 8 bytes
```

The above errors can also be related to encrypted data which has been encrypted with different merchant keys. In that specific case please refer to WebSphere Commerce V5.6 migration guide.

Note: Encrypted data is identified with a space at the front of the value. If columns exists in the targeted tables with values that begin with space and is not encrypted the above error will occur. If such values exists they must be corrected by trimming the space(s) in the beginning of the value.

To see the complete migration script containing all the above steps, please refer to Appendix C, "Migration scripts" on page 255.

11.4 Migrating Payment Manager

In the following migration scenario, WebSphere Payment Manager Node is the old system to be migrated and WebSphere Commerce Payments Node is the new system.

The following list highlights the necessary steps for the remote payment migration:

1. Ensure that WebSphere Payment Manager is the right version. Supported versions are 2.2.1.x, 3.1.3.x and 5.5.0.0.
2. Install WebSphere Commerce Payments V5.6 on the WebSphere Commerce Payments Node.

3. On the WebSphere Commerce Payments Node, run WCIM to package the tool.
4. Transfer WCIM tool package to the WebSphere Payment Manager Node.
5. On the WebSphere Payment Manager Node, run WCIM to backup instance and other assets.
6. Backup WebSphere Payment Manager database.
7. Transfer the backups to the WebSphere Commerce Payments Node.
8. Restore WebSphere Payment Manager database on the WebSphere Commerce Payments Node.
9. Run the WCIM tool on the WebSphere Commerce Payments Node to migrate the instance
10. Migrate the payments database.

11.4.1 Install Fixpack for WebSphere Payment Manager

If WebSphere Payment Manager version is earlier to 2.2.1, apply the necessary fixes to get it to the right level, refer to the WebSphere Payment Manager support site at the following URL:

<http://www.ibm.com/software/genservers/commerce/paymentmanager/support.html>

Do the following on the WebSphere Payment Manager Node:

1. Download the fixpack file,
`IBMPayMgr_nt_Framework_Ctr_2_2_1_0_ptf.class`
to a temporary directory.

Note: The fixpack installer creates several temporary files in this folder, so we recommend that you use an empty directory for storing and executing the fixpack.

2. Stop both WebSphere Payment Manager and WebSphere Application Server.

To stop the WebSphere Payment Manager run the following command from the <WPM_home>\ directory:

```
stopIBMPayServer
```

In our example

```
D:\WebSphere\PaymentManager
```

To stop the WebSphere Payment Manager application server follow these steps:

- a. Open the WebSphere Application Server Administrative Console
 - b. Expand the Administrative Domain
 - c. Expand the Node
 - d. Right click on the WebSphere Payment Manager application server and select **Stop**
 - e. Click **OK** on the information dialog window that confirms the command completed successfully
 - f. Close the WebSphere Application Server Administrative Console
3. Issue the following commands in a Windows command window:

```
SET PATH=<was_home>\jdk\jre\bin;%PATH%
java -cp . IBMPayMgr_nt_Framework_Ctr_2_2_1_0_ptf
```

In our example, we used:

```
SET PATH=D:\WebSphere\Appserver\jdk\jre\bin;%PATH%
java -cp . IBMPayMgr_nt_Framework_Ctr_2_2_1_0_ptf
```

4. The welcome window will appear. Click **Next**.
5. The Enter Database Password window appears. Enter the password for the database administrator and click **Next**.
6. The Verify Publish Directory window appears. Verify that the directory shown corresponds to the Web Server root for the WebSphere Payment Manager files. If this is not the case, click **Browse** and select the correct directory. Click **Next** when done.
7. The fixpack installation will complete. After it has completed, a confirmation window will appear. Uncheck **Display the Readme file** and click **Next**.
8. After the installation is complete, you should verify the contents of the following two files that are created in the directory from which the fixpack installer is executed:

```
default.trace
trace_FRAME_ptf.trace
```

Ensure that none of these files contain any error messages.

11.4.2 Install WebSphere Commerce Payments remote node

In Chapter 9, “Installing WebSphere Commerce V5.6” on page 163 we covered the steps to install WebSphere Commerce V5.6 components on a single node environment. The following instructions provide the necessary steps to install WebSphere Commerce Payments V5.6 on a remote node.

1. Insert the WebSphere Commerce V5.6 CD 1 and run **setup.exe**

2. On the Language selection window, select the language and click **OK**
3. The Welcome window will appear. Click **Next**.
4. Select **I accept the terms in the license agreement** and click **Next**.
5. The Installation Wizard will display a warning window. Check to see if these warnings apply to you. Click **Next** when all problems have been resolved.
6. The system will prompt for the installation type. Select **Custom Installation** and click **Next**.
7. Select the following components:
 - WebSphere Commerce Payments
 - IBM DB2 Universal Database Enterprise Server Edition
 - IBM HTTP Server
8. The database and Web server selection window appears. Since we elected to install IBM DB2 and IBM HTTP Server in the previous window, it is not possible to modify the drop-down boxes. Click **Next**
9. The destination paths window appears. Enter the desired destination paths for the components being installed and click **Next**

We used the following values:

 - IBM DB2 Universal Database: D:\WebSphere\SQLLIB
 - IBM HTTP Server: D:\WebSphere\HTTPServer
 - IBM WebSphere Application Server: D:\WebSphere\AppServer
 - IBM WebSphere Commerce Server: D:\WebSphere\CommerceServer
10. You are prompted to enter the user ID and password for the administrative user. Enter the user ID and password for the current Windows user and click **Next**
11. A confirmation message is shown, acknowledging that the user entered has the desired user privileges. Click **OK**
12. The installation options confirmation window appears. Check the information and click **Next**
13. Insert the IBM DB2 Universal Database V8.1.5 CD when the Installation Wizard prompts for it and click **Next**
14. Insert the IBM WebSphere Application Server V5.0 CD when the Installation Wizard prompts for it and click **Next**
15. Insert the IBM WebSphere Application Server Fixpack CD when the Installation Wizard prompts for it and click **Next**
16. Insert the IBM WebSphere Commerce V5.6 CD 1 when the Installation Wizard prompts for it and click **Next**

17. Insert the IBM WebSphere Commerce V5.6 CD 2 when the Installation Wizard prompts for it and click **OK**
18. A window confirming that WebSphere Commerce V5.6 has been installed is shown. Click **Next**
19. The Installation Wizard prompts for restarting the system. Ensure that **Restart now** is selected and click **Finish**

11.4.3 Package and transfer the WCIM tool

It is necessary to create a package with the WCIM tool in the WebSphere Commerce Payments Node and transfer it to WebSphere Payment Manager Node in order to have this functionality in the original node.

1. On the WebSphere Commerce Payments Node, run WCIM to package the tool. In `<wc_home>\bin`, run:

```
wcim -component wcim
```

This will produce a file `<wc_home>\temp\zip\backupwcim.zip` containing the necessary files to run the WCIM tool on the WebSphere Payment Manager Node.

Note: At the time of writing this redbook, we found that in order to package the WCIM tool, you would need the following files from a WebSphere Commerce V5.6 installation:

- ▶ `<wc_home>\xml\migration\backupwcim.xml`
- ▶ `<wc_home>\xml\migration\instbackupwcp313.xml`
- ▶ `<wc_home>\xml\migration\instbackupwcp55.xml`
- ▶ `<wc_home>\xml\migration\instbackupwcp56up.xml`
- ▶ `<wc_home>\xml\migration\instbackupwpm221.xml`
- ▶ `<wc_home>\xml\migration\instbackupwpm312.xml`

The error message from WCIM when these files are missing would be:

```
Error: File C:\WEBSPH~1\COMMER~1\xml\migration\backupwcim.xml does not exist. Terminate WCIM.
```

Where `C:\WEBSPH~1\COMMER~1` is the Windows short name for our `<wc_home>` directory.

Copy these files from your WebSphere Commerce V5.6 node if they are not present on your WebSphere Commerce Payments V5.6 node before running the above command.

2. Copy the file `<wc_home>\temp\zip\backupwcim.zip` to the WebSphere Payment Manager Node directory `<wpm_home>\temp\zip` and unzip it to the directory `<wpm_home>\temp`.

For example, we unzipped the file to the directory

`D:\WebSphere\PaymentManager\temp`, creating the following subdirectories:

- `bin`
- `java`
- `lib`
- `payments`
- `xml`

11.4.4 Backup the WebSphere Payment Manager instance

Use the WCIM tool to backup the WebSphere Payment Manager instance.

On the WebSphere Payment Manager Node, type the following command from the `<wpm_home>\temp\bin` directory:

```
wcim -component wpm -from wpm221 -backup remote -instance default
```

When we ran the command, it prompted us for the following values:

- The directory path where the temp directory reside
This is the location in which you unzipped the **backupwcim.zip** file, `<wpm_home>\temp`.
In our example, we entered `D:\WebSphere\PaymentManager`.
- Old Payment home directory
This is the main installation directory for WebSphere Payment Manager, `<wpm_home>`.
In our example, we entered `D:\WebSphere\PaymentManager`.
- The previous IBM HTTP Server home directory
This is the document root of the locally installed Web server, also known as `htdocs`. For a standard installation, it will be `<ihs_home>\htdocs`.
In our example, we entered `D:\WebSphere\HTTP\htdocs`.

Note: At the time of writing this book WebSphere Payment Manager instance backup failed because of a missing file, the error shows as follows:

```
Error: File C:\Program Files\IBM\PaymentManager\temp\xml\migration\
instbackupwpm221.xml does not exist. Terminate WCIM.
```

Copy this file from the WebSphere Commerce V5.6 node if it is not present on your WebSphere Commerce Payments Node before running the above command.

After the script runs, a file **wcbbackupwpm221.zip** will be created in the directory <wpm_home>\temp\zip.

Note: The issues mentioned here are those we encountered while running the script. Depending on your configuration, you may encounter different issues when you run the script in your environment.

11.4.5 Backup the WebSphere Payment Manager database

To backup the existing WebSphere Payment Manager database do the following on the WebSphere Payment Manager Node:

1. Ensure that WebSphere Payment Manager and WebSphere Application Server are not running
2. Create a backup directory. For example, we created the following directory:
c:\db2backup
3. Close all database connections by typing the following command from a DB2 command line:

```
DB2 force application all
```

4. Backup the payment database

```
DB2 backup db <wpm_db> to <backup_dir>
```

Where <wpm_db> is the name of your WebSphere Payment Manager database and <backup_dir> is the backup directory that you created in step 2.

Review the file <wpm_home>\IBMPayServer.cmd if you are not sure about the name of your WebSphere Payment Manager database. The name of the database will show in the parameter DBjdbcURL, as jdbc:db2:<wpm_db>, to the com.ibm.etill.framework.ETill class.

In our example:

```
DB2 backup db payman to c:\db2backup
```

11.4.6 Transfer assets to WebSphere Commerce Payments node

The assets need to be transferred from the WebSphere Payment Manager Node to the new WebSphere Commerce Payments Node.

1. Transfer the instance backup file, **wcbakupwpm221.zip**, from the <wpm_home>\temp\zip directory on the WebSphere Payment Manager Node to the <wcp_home>\temp\zip directory on the WebSphere Commerce Payments node.
2. Transfer the WebSphere Payment Manager database backup, you made in 11.4.5, “Backup the WebSphere Payment Manager database” on page 223, to the WebSphere Commerce Payments Node.

11.4.7 Restore WebSphere Payment Manager database

Restore the WebSphere Payment Manager database on the WebSphere Commerce Payments Node by typing the following command from the DB2 Command Prompt:

```
db2 restore db <wpm_db> from <backup_dir>
```

In our example:

```
db2 restore db payman from c:\db2backup
```

11.4.8 Migrate the WebSphere Payment Manager instance

To migrate the WebSphere Payment Manager instance follow these steps on the WebSphere Commerce Payments node.

1. Type the following command from the <Commerce_home>\bin directory

```
wcim.bat -component wpm -from wpm221 -migration replace  
-instance <payment_instance_name>
```

In our example we typed

```
wcim.bat -component wpm -from wpm221 -migration replace -instance default
```

When we ran the command, it prompted us for the following values:

- Is a new Web server host name used for the new instance? [DEFAULT yes]

In our example, we entered **YES**

- Enter a new Web server host name:[DEFAULT <full_qualified_hostname>]

In our example, we accepted the **default**

- Please type the new WAS node name: [DEFAULT <hostname>]
In our example, we accepted the **default**
 - Please type Payment DataBase password:
In our example, we entered dbuser01pwd
2. Verify the logs located in <Commerce_home>\temp\logs and review in the following file:
`wcim.migration.wpm.default.wpm221.<timestamp>.log`
 look for the following entry at the end of the file
`[<timestamp>] Event: WCIM has completed the job(s) assigned successfully.`
 3. Update WebSphere Application Server settings following these steps:
 - a. from the command prompt, change to the <WAS_home>\bin directory and start the server1 by typing the following command:
`startserver server1`
 - b. logon to the WebSphere Application Server Administration Console by pointing your browser to the following URL:
`http://<hostname>:9090/admin`
 - c. type a User ID and click **OK**
 - d. on the left panel, expand **Environment** and click on **Virtual Hosts** and then on **VH_PYM_mig_<payment_instance>**
 In our example
`VH_PYM_mig_default`
 - e. in the Additional Properties section click on **Host Aliases**
 - f. for each host name listed, click on the <host name> and verify that the port specified is **5433** and click on **OK**
 - g. on the left panel, expand **Resources**, click on **JDBC Providers** and then click on the **New** button
 - h. select **DB2 Legacy CLI-based Type 2 JDBC Driver** from the JDBC Providers pull-down menu and click **OK**
 - i. on the Configuration page specify the classpath for the DB2 java classes
 In our example:
`D:\WebSphere\SQLLIB\Java\db2java.zip`
 - j. click on **OK**
 - k. click on the **DB2 Legacy CLI-based Type 2 JDBC Driver** link, scroll down and then click on **Data Sources (Version 4)**

- l. click on **New** to create a new datasource
 - m. we used these values to create the new datasource:
 - **Name:** migrated_default Commerce Payments Datasource
 - **JNDI Name:** jdbc/ migrated_default Commerce Payments Datasource
 - **Database name:** payman
 - **Default User ID:** dbusr01
 - **Default Password:** dbusr01pwd
 - click **OK**
 - n. in the messages area, click on **Save** and then again on the **Save** button to save these changes
 - o. Go back to the **JDBC Datasource** and test the connection
 - p. click on **Logout** and close the Web browser
 4. Update the Configuration Manager settings by following these steps:
 - a. start the **WC 5.6 Configuration Manager** service
 - a. start the **WebSphere Commerce Configuration Manager** and log in
 - b. expand the entries until **Instance Properties** (at this point you should have just one instance)
 - c. click on **WebServer**
 - d. change the **Server Port** to **5433**
 - e. check the **Use SSL** check box
 - f. click **Apply**
 - g. click **OK** on the following two windows
 - h. click on **WCSRealm**
 - i. change the **Commerce Webserver Hostname** to the WebSphere Commerce V5.6 hostname
 - j. change the **WC Web Server Port** to **5433**
 - k. click **Apply**
 - l. click **OK** on the following confirmation window
 - m. close the **WebSphere Commerce Configuration Manager**
 - n. restart the Web server

Note: When configuring these parameters pay special attention to these values, they must match to what you have configured in the commerce instance. Make sure that the commerce instance has properly configured the WebSphere Commerce Payments Web Server Hostname and ports values.

11.4.9 Migrate the WebSphere Payment Manager database

Before migrating the payments database there are some records that must be deleted from both payments and commerce databases. Connect to the commerce database and delete the obsolete payment methods from the PAYMTHD table. They are no longer supported.

1. On the database node connect to the commerce database by typing the following commands from a DB2 command line:

```
DB2 connect to <WC_db> using <db_user_id> using <db_user_password>
```

In our example:

```
DB2 connect to mall using dbusr01 using dbusr01pwd
```

2. Delete the obsolete payment methods:

```
db2 delete from paymthd where profilename in  
( 'WCS51_SET_MIA', 'WCS51_SET_Wallet', 'WCS51_CyberCash' )
```

3. Disconnect from the commerce database:

```
DB2 connect reset
```

Move to the WebSphere Commerce Payments Node and delete the keys from the ETKEY table. They will be recreated during the migration.

1. On the WebSphere Commerce Payments Node connect to the payments database by typing the following commands from a DB2 command line:

```
DB2 connect to <WCP_db>
```

In our example:

```
DB2 connect to payman
```

2. Delete the obsolete payment methods:

```
db2 delete from ETKEY
```

3. Disconnect from the payments database:

```
DB2 connect reset
```

Now you can proceed with the payments database migration by following these steps:

1. Still on the WebSphere Commerce Payments Node open a DB2 Command Window and navigate to the <Commerce_home>\bin directory, in our example:

D:\WebSphere\CommerceServer56\bin

2. run migratepaymentsdb.bat with the following parameters:

migratepaymentsdb db2 <db_name> <user_ID> <password> <db_schema>

in our example

migratepaymentsdb db2 payman dbusr01 dbusr01pwd dbusr01

This concludes the payments database migration.



Post-migration steps

This chapter describes the additional steps to be performed after the production instance and database have been migrated.

This chapter is organized in the following sections:

- ▶ Post migration steps for IBM HTTP Server
 - Migrating static content
 - Updating configuration
- ▶ Deploying
 - Deploying EJBs
 - Deploying commands and databeans
 - Deploying store assets

12.1 Post migration steps for IBM HTTP Server

After migrating WebSphere Commerce, there are still some steps that require to be done manually. These steps will vary from one environment to another depending if it is a single or a multi-node environment.

12.1.1 Migrating static content

The static HTML content must be moved over manually. This is done simply by copying all the content from the HTTP document root on WebSphere Commerce Suite V5.1 system to the HTTP document root on WebSphere Commerce V5.6 system. In our example we had our document root set to:

```
d:\static_http
```

This directory was copied from WebSphere Commerce Suite V5.1 system to WebSphere Commerce V5.6 system.

Note: This should be enough as the document root will already be set to the correct directory (IBM HTTP Server settings are preserved, see Updating configuration)

If the document root is not set, it can be set manually in the IBM HTTP Server configuration file. The configuration of the IBM HTTP Server is stored in:

```
<ihs_home>\conf\httpd.conf
```

in our example:

```
D:\WebSphere\IBMHttpServer\conf\httpd.conf
```

The document root is added as described below:

```
DocumentRoot "d:/static_http/content/htdocs"
```

The IBM HTTP Server needs to be restarted in order for the changes to take effect.

12.1.2 Updating configuration

The WCIM tool will migrate the IBM HTTP Server configuration automatically, if IBM HTTP Server is on WebSphere Commerce V5.6 system, installed locally on the same node. All customized configuration in the configuration file (**httpd.conf**) will be migrated, assuming that no customized configuration exists in the blocks tagged as shown in Example 12-1.

Example 12-1 WebSphere Commerce section of httpd.conf

```
##### IBM WebSphere Commerce (Do not edit this section) #####
.
.
.
##### End of IBM WebSphere Commerce (Do not edit this section) #####
```

If one of the following Web Server is used, then the configuration must be migrated manually:

- ▶ IBM HTTP Server running remote from WebSphere Commerce V5.6
- ▶ Sun ONE Web Server (formerly known as iPlanet Web Server) running local or remote from WebSphere Commerce V5.6

Note: The Lotus Domino Web Server is no longer supported.

Refer to *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1* for details on how to configure the above mentioned Web servers.

12.2 Deploying

In order to finish the migration we must do a fresh deploy to the runtime environment. This deploy will contain a build of all migrated EJBs, commands and databeans and store assets. In each section below is described how to deploy the individual components of the complete build.

The type of deployment described in this section is an incremental deployment. In an incremental deployment, you must already have a WebSphere Commerce enterprise application installed on the production WebSphere Commerce Server.

12.2.1 Deploying EJBs

Deploying the EJB consists of 2 steps:

1. Configuring and exporting the JAR file
2. Applying the JAR file to the production environment

The steps below describes the steps needed to deploy EJBs. These steps are only valid for DB2 in development and production environment. If the production database is different to the development database, refer to WebSphere Commerce V5.6 Information Center.

Configure and export the JAR file

To configure and export the JAR file, perform the following steps:

1. Open WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment > WebSphere Commerce development environment**) and switch to the Project Navigator view.
2. Expand the EJB project **WebSphereCommerceServerExtensionsData**
3. Double-click **EJB Deployment Descriptor**.
4. With the Overview tab selected, scroll to the bottom of the pane, to locate the WebSphere Bindings section.
5. In the DataSource JNDI name field, enter the data source JNDI name of the runtime WebSphere Commerce V5.6.

Note: The value of the JNDI datasource can be found in the WebSphere Commerce V5.6 <instance.xml> file and is not necessarily the same as the one in WebSphere Commerce V5.6 development environment. In the <instance.xml> file the key **DatasourceName** is used to specify the JNDI datasource name.

In our example the JNDI datasource name was 'WebSphere Commerce Suite DB2 DataSource demo'

6. Save your deployment descriptor changes (Ctrl+S) and close it.
7. In the Project Navigator view, right-click the EJB project and select Export. The Export wizard opens.
8. In the Export wizard, do the following:
 - a. Select EJB JAR file and click Next.
 - b. The EJB project name is prepopulated. Leave this value as is.
 - c. For the destination, enter **WebSphereCommerceServerExtensionsData.jar**.
 - d. Ensure that Export source files is not selected.
 - e. Click Finish.
9. After the JAR file has been created, open the EJB deployment descriptor and restore the modifications that were made in step 5, back to the setting that is required for your local test server. Save your changes.

Applying the JAR file to the production environment

To apply the JAR file to the production environment, perform the following steps:

1. Stop the WebSphere Commerce instance that is running within WebSphere Application Server V5.0.2.
2. Locate the original EJB JAR file (**WebSphereCommerceServerExtensionsData.jar**) in the WebSphere Commerce instance. The EJB file will be located in:
`<was_home>\installedApps\host_name\instance_name.ear`
3. Make a backup copy of the original EJB JAR file to a convenient directory
4. Overwrite original EJB JAR file on the production machine with the new file from the development machine
5. If the **EJB deployment descriptor has been modified**, perform the following steps:
 - a. Locate the deployment repository (**META-INF** directory) on the production machine. The META-INF directory can be found here:
`<was_home>\config\cells\hostname
 \applications\instance_name.ear\deployments\
 instance_name\WebSphereCommerceServerExtensionsData.jar\META-I
 NF`
 - b. Backup all of the files in the directory to a convenient directory
 - c. Use a tool to open the **WebSphereCommerceServerExtensionsData.jar** file and view its contents.
 - d. Extract the contents of the META-INF directory from the **WebSphereCommerceServerExtensionsData.jar** file into the directory from step 5a.
6. Restart the WebSphere Commerce instance.

12.2.2 Deploying commands and databeans

Deploying the commands and databeans consists of 2 steps:

1. Exporting the JAR file
2. Applying the JAR file to the production environment

Export the JAR file

To export the commands and databeans perform the following steps:

1. Open WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment > WebSphere Commerce development environment**) and switch to the Project Navigator view

2. Right-click the **WebSphereCommerceServerExtensionsLogic** project and select **Export**.

The Export wizard opens.

3. In the Export wizard, do the following:
 - a. Select **JAR file** and click **Next**.
 - b. Ensure that **Export generated class files and resources** is selected.
 - c. Do not select **Export Java source files and resources**.
 - d. In the **Select the export destination** field, enter **WebSphereCommerceServerExtensionsLogic.jar** as the JAR file name to use
 - e. Click **Finish**.

Applying the JAR file to the production environment

To apply the JAR file to the production environment perform the following:

1. Stop the WebSphere Commerce instance that is running within WebSphere Application Server
2. Locate the original JAR file (**WebSphereCommerceServerExtensionsLogic.jar**) in the WebSphere Commerce instance. The JAR file will be located in:
`<was_home>\installedApps\host_name\instance_name.ear`
3. Make a backup copy of the original JAR file to a convenient directory
4. Overwrite original JAR file on the production machine with the new file from the development machine
5. Restart the WebSphere Commerce instance.

Note: Before starting the WebSphere Commerce instance, apply any necessary change to the database, such as command registration, view registration, additional tables, access control polices, etc.

12.2.3 Deploying store assets

Deploying the store assets consists of 2 steps:

1. Exporting store assets
2. Applying store assets to the production environment

Exporting store assets

To export store assets perform the following steps:

1. Open WebSphere Commerce development environment (**Start > Programs > IBM WebSphere Commerce development environment > WebSphere Commerce development environment**) and switch to the Project Navigator view
2. Expand the **Stores** folder
3. Right-click the **Web Content** directory and select **Export**.
The Export wizard opens.
4. In the Export wizard, do the following:
 - a. Select **File system** and click **next**
 - b. Select all of the resources that you want to deploy. Resources include JSP files, HTML files, images, property files and other store assets.
 - c. Select **Create directory structure for files**
 - d. In the **Directory** field, enter a temporary directory into which these resources will be placed.
 - e. Click **Finish**

Applying the store assets to the production environment

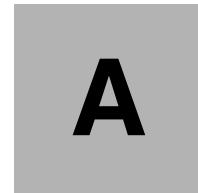
To apply the store assets perform the following steps:

1. Depending upon the types of assets you are deploying and your specific configuration details, you may need to stop the WebSphere Commerce instance that is running within WebSphere Application Server V5.0.2.
2. Copy the store assets exported into:
`<was_home>\installedApps\host_name\instance_name.ear\Stores.war`
3. If you previously stopped the WebSphere Commerce instance, start it.



Part 4

Appendixes



Managing WebSphere Commerce components

The following appendix provides information on how to check the status, start and stop various WebSphere Commerce components for the runtime environment as well as detailed information for enabling tracing and re-creating missing password scripts for the development environment.

This appendix is organized in the following sections

For runtime environment:

- ▶ WebSphere Commerce instance management
- ▶ WebSphere Commerce Payments instance management
- ▶ Start the Configuration Manager
- ▶ Restarting IBM HTTP Server
- ▶ Modifying trace strings for a running server

For development environment:

- ▶ Tracing with WebSphere Commerce V5.6 Toolkit
- ▶ Create missing password scripts

WebSphere Commerce instance management

This section describes how to start and stop, as well as how to check the status, of the WebSphere Commerce instance.

Note: The parameters to the serverStatus, startServer and stopServer scripts are case sensitive.

Checking instance status

Run the following command from the <was_home>\bin directory:

```
serverStatus WC_<instancename>
```

Where <instancename> is the name of the instance. If the instance is running, the last message from the serverstatus command will be:

```
ADM0508I: The Application Server "WC_demo" is STARTED
```

If the instance is not running, the last message will be:

```
ADMU0509I: The Application Server "WC_<instancename>" cannot be reached. It appears to be stopped.
```

Other messages are possible, for example when the instance is starting or stopping.

Starting the instance

Run the following command from the <was_home>\bin directory:

```
startServer WC_<instancename>
```

Where <instancename> is the name of the instance. When the instance is started, the last message from the serverstatus command should be:

```
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server WC_<instancename> open for e-business; process id is  
<process_id>
```

If the instance was already running, the some of the messages from the startServer command will be:

```
ADMU3028I: Conflict detected on port 8882. Likely causes: a) An instance  
of the server WC_<instancename> is already running b) some other process  
is using port 8882  
ADMU3027E: An instance of the server may already be running:  
WC_<instancename>
```

```
ADMU0111E: Program exiting with error:  
com.ibm.websphere.management.exception.AdminException: ADMU3027E: An  
instance of the server may already be running: WC_<instancename>
```

Stopping the instance

Run the following command from the <was_home>\bin directory:

```
stopServer WC_<instancename>
```

Where <instancename> is the name of the instance. When the instance is stopped, the last message from the serverstatus command should be:

```
ADMU3201I: Server stop request issued. Waiting for stop status.  
ADMU4000I: Server WC_<instancename> stop completed.
```

If the instance is not running, the last message will be:

```
ADMU0509I: The Application Server "WC_<instancename>" cannot be reached. It  
appears to be stopped.
```

WebSphere Commerce Payments instance management

This section describes how to start and stop, as well as how to check the status, of the WebSphere Commerce Payments instance.

Note: The parameters to the serverStatus, startServer and stopServer scripts are case sensitive.

Checking instance status

Run the following command from the <was_home>\bin directory:

```
serverStatus <instancename>_Commerce_Payments_Server
```

Where <instancename> is the name of the instance. If the instance is running, the last message from the serverstatus command will be:

```
ADM0508I: The Application Server "<instancename>_Commerce_Payments_Server"  
is STARTED
```

If the instance is not running, the last message will be:

```
ADMU0509I: The Application Server  
"<instancename>_Commerce_Payments_Server" cannot be reached. It appears to  
be stopped.
```

Other messages are possible, for example when the instance is starting or stopping.

Starting the instance

Run the following command from the <was_home>\bin directory:

```
startServer <instancename>_Commerce_Payments_Server
```

Where <instancename> is the name of the instance. When the instance is started, the last message from the serverstatus command should be:

```
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server <instancename>_Commerce_Payments_Server open for  
e-business; process id is <process_id>
```

If the instance was already running, the some of the messages from the startServer command will be:

```
ADMU3028I: Conflict detected on port 8882. Likely causes: a) An instance  
of the server <instancename>_Commerce_Payments_Server is already running  
b) some other process is using port 8882  
ADMU3027E: An instance of the server may already be running:  
<instancename>_Commerce_Payments_Server  
ADMU0111E: Program exiting with error:  
com.ibm.websphere.management.exception.AdminException: ADMU3027E: An  
instance of the server may already be running:  
<instancename>_Commerce_Payments_Server
```

Stopping the instance

Run the following command from the <was_home>\bin directory:

```
stopServer <instancename>_Commerce_Payments_Server
```

Where <instancename> is the name of the instance. When the instance is stopped, the last message from the serverstatus command should be:

```
ADMU3201I: Server stop request issued. Waiting for stop status.  
ADMU4000I: Server <instancename>_Commerce_Payments_Server stop completed.
```

If the instance is not running, the last message will be:

```
ADMU0509I: The Application Server "<instancename>_Commerce_Payments_Server"  
cannot be reached. It appears to be stopped.
```

Start the Configuration Manager

Follow these steps to start the Configuration Manager:

1. Ensure that the DB2 Universal Database V8.1 instance is running by entering the following command in a Windows command window:

db2start

The output from this command should be as follows:

```
SQL1063N  DB2START processing was successful.
```

If the instance was already running, the output will be as follows:

```
SQL1026N  The database manager is already active.
```

2. Start the Configuration Manager Server by entering the following command in a Windows command line:

net start "IBM WC 5.6 Configuration Manager"

The output from this command should be as follows:

```
The IBM WC 5.6 Configuration Manager service is starting..
```

```
The IBM WC 5.6 Configuration Manager service was started successfully.
```

If the service was already running, the output will be as follows:

```
The requested service has already been started.
```

```
More help is available by typing NET HELPMSG 2182.
```

Tip: The Configuration Manager Server can also be started from the Windows services window.

3. Select **Start -> IBM WebSphere -> Commerce Server 5.6 -> Configuration** to start the Configuration Manager Client.
4. The Config Authentication window will appear. Enter the user ID and password for the Configuration Manager using the following values and click **OK**:

User ID: webadmin
Password: webibm
5. A message, prompting you to change the password is shown. Click **OK**.
6. The Modify Password window will appear. Choose a new password for the webadmin user and enter it in the **New Password** and **Confirm New Password** fields and click **OK**. Note that the new password must adhere to the following rules:
 - Must contain at least one alphabetic character.
 - Must contain at least one numeric character.
 - The same character cannot occur more than three times.

Tip: If you forget the password for the webadmin user, you can reset the password manually by editing the file <wc_home>\instances\PwdMgr.xml. This file contains the password in encrypted form. To change the password, run the command:

```
wcs_encrypt <new_password>
```

Where <new_password> is the new password for the webadmin user. Copy the ASCII encrypted string output from this command and insert it into the LoginPassword node of the PwdMgr.xml file.

7. After changing the password, the Configuration Manager opens.

Restarting IBM HTTP Server

During the instance creation, the Commerce Instance Creation Wizard may instruct you to restart the Web server. The following instructions show how to restart the IBM HTTP Server:

1. Open a Windows command prompt.
2. Issue the following three commands:

```
net stop "IBM HTTP Server 1.3.26"  
net start "IBM HTTP Server 1.3.26"
```

The output should be similar to what is shown in Example A-1.

Example: A-1 Restarting the IBM HTTP Server

```
C:\>net stop "IBM HTTP Server 1.3.26"  
The IBM HTTP Server 1.3.26 service is stopping....  
The IBM HTTP Server 1.3.26 service was stopped successfully.
```

```
C:\>net start "IBM HTTP Server 1.3.26"  
The IBM HTTP Server 1.3.26 service is starting....  
The IBM HTTP Server 1.3.26 service was started successfully.
```

Modifying trace strings for a running server

When changing the trace specification via the WebSphere Application Server V5.0.2 Administration Console, the changes does not take effect until the server is restarted.

Follow the instructions in this section to modify the trace specification for a running server.

Note: The commands shown in this section are very long and may span more than one line. All commands should be entered on one single line in the wsadmin tool.

1. Identify the SOAP port for your server by opening the **SystemOut.log** file, residing in the directory <was_home>\logs\<servername>. Search for the string SOAP connector available at port. The line in the log file should look like the following:

```
[<date> <time>] <threadId> JMXSoapAdapte A ADMC0013I: SOAP connector
available at port <portnumber>
```

Where <portnumber> is the SOAP port for your server. <date>, <time> and <threadId> are variables, depending on the time your server was started.

In our example, we wanted to change the trace specification for the WebSphere Commerce instance named demo, so we found the following line in the file D:\WebSphere\AppServer\logs\WC_demo\SystemOut.log:

```
[6/1/04 19:08:01:190 EDT] 4df9f657 JMXSoapAdapte A ADMC0013I: SOAP
connector available at port 8881
```

Thus, the SOAP port number in our example is 8881.

Tip: The SOAP port for the server1 server is always 8880, which is the default for the wsadmin tool. If you need to change the trace string for server1 you will thus not need to specify any parameters to wsadmin when launching it.

2. Start the wsadmin tool against your server by issuing the following command from the <was_home>\bin directory:

```
wsadmin -conntype SOAP -port <portnumber>
```

Where <portnumber> is the number identified in step 1. The wsadmin tool will launch with the following messages. The last line is the wsadmin prompt:

```
WASX7209I: Connected to process "<servername>" on node <nodename> using
SOAP connector; The type of process is: <processtype>
WASX7029I: For help, enter: "$Help help"
wsadmin>
```

In our example, we issued the following command:

```
wsadmin -conntype SOAP -port 8881
```

Resulting in the following messages:

```

WASX7209I: Connected to process "WC_demo" on node WC56 using SOAP
connector; The type of process is: UnManagedProcess
WASX7029I: For help, enter: "$Help help"
wsadmin>

```

Note: If you get the following messages when attempting to launch wsadmin, you have either specified a wrong SOAP port number, or the server you are trying to reach is not started:

```

WASX7023E: Error creating "SOAP" connection to host "<hostname>";
exception information:
com.ibm.websphere.management.exception.ConnectorNotAvailableException
WASX7213I: This scripting client is not connected to a server process;
please refer to the log file <was_home>\logs\wsadmin.traceout for
additional information.
WASX7029I: For help, enter: "$Help help"
wsadmin>

```

Where <hostname> and <was_home> are the host name of your server, and the installation directory for WebSphere Application Server V5.0.2, respectively.

In this case, you must exit wsadmin and remedy the situation. To exit from wsadmin, enter the following at the wsadmin prompt:

```
quit
```

3. Optional: Issue the following command as the wsadmin prompt to check the current trace string for your server:

```
$AdminControl getAttribute [$AdminControl completeObjectName
type=TraceService,*] traceSpecification
```

The response from wsadmin is simply a line containing the trace setting, for example:

```
*=all=disabled
```

4. To modify the trace string, issue the following command at the wsadmin prompt:

```
$AdminControl setAttribute [$AdminControl completeObjectName
type=TraceService,*] traceSpecification <traceString>
```

Where <traceString> is the desired trace string.

In our example, we wished to enable tracing for the WebSphere Commerce component ORDER, so we issued the following command:

```
$AdminControl setAttribute [$AdminControl completeObjectName
type=TraceService,*] traceSpecification
com.ibm.websphere.commerce.WC_ORDER=all=enabled
```

Tip: The trace strings for the various WebSphere Commerce components can be found in the WebSphere Commerce InfoCenter at the following address:

<http://publib.boulder.ibm.com/infocenter/wc56help/topic/com.ibm.commerce.admin.doc/refs/rlslogging.htm>

Also, to enable logging for more than one component, separate the trace strings with a colon, for example:

```
com.ibm.websphere.commerce.WC_ORDER=all=enabled:com.ibm.websphere.commerce.WC_INVENTORY=all=enabled:com.ibm.websphere.commerce.WC_UTF=all=enabled
```

This will enable tracing for the components ORDER, INVENTORY and UTF.

5. Enter the following at the wsadmin prompt to exit the wsadmin tool:

```
quit
```

Note: The changes to the trace string are not saved in the server configuration. If the server is restarted, the trace string specification will revert to the value it had before you changed the value from within the wsadmin tool.

To permanently change the trace string for a server, you must use the WebSphere Commerce Administration Console.

Tracing with WebSphere Commerce V5.6 Toolkit

Specifying trace levels and components for WebSphere Commerce V5.6 is very different than in WebSphere Commerce Suite V5.1. Since WebSphere Commerce V5.5, the WebSphere Application Server tracing facility has been used for trace output.

In the WebSphere Commerce V5.6 runtime environment, tracing is enabled in the WebSphere Application Server Administration Console.

In WebSphere Commerce V5.6 Toolkit, the tracing is specified in one of two ways, depending on whether the Full or the Lightweight WebSphere Test Environment is used.

Tracing for Lightweight WebSphere Test Environment

The Lightweight WebSphere Test Environment uses a property file to control which components have tracing enabled.

To change the tracing level and which components have tracing enables, edit the following file:

```
<wctoolkit_home>\properties\com\ibm\commerce\litecontainer\Logging.properties.
```

Where <wctoolkit_home> is the base directory for the WebSphere Commerce V5.6 Toolkit. Uncomment the components that you wish to enable tracing for. Example A-2 shows a part of this file with the tracing enabled for the component EXTERN and trace level raised to capture warning and informational messages as well. All modified lines have been emphasized in bold.

Example: A-2 Excerpt from Logging.properties file

```
# logging settings
LOG_ERROR=true
LOG_WARNING=true
LOG_INFORMATION=true

# uncomment components that you want to trace

#com.ibm.websphere.commerce.WC_ACCESSCONTROL=true
#com.ibm.websphere.commerce.WC_AC_UNITTEST=true
#com.ibm.websphere.commerce.WC_APPROVAL=true
#com.ibm.websphere.commerce.WC_BI=true
#com.ibm.websphere.commerce.WC_CATALOG=true
:
:
:
com.ibm.websphere.commerce.WC_EXTERN=true
:
:
:
#com.ibm.websphere.commerce.WC_UBF=true
#com.ibm.websphere.commerce.WC_UTF=true
#com.ibm.websphere.commerce.payments.MPF=true
#com.ibm.websphere.commerce.payments.OfflineCard=true
```

The trace output is shown in the **Console** panel and is stored in the file <wctoolkit_home>\AppServer\logs\server1\trace.log.

Tracing for Full WebSphere Test Environment

To change the trace settings for the Full WebSphere Test Environment, you need to edit the server configuration from within WebSphere Studio Application Developer V5.1.1:

1. Double-click the **Full WebSphere Test Environment** server in the **Servers** panel of the **Debug** or **J2EE** perspectives.
2. Expand the **Tracing** section on the **Server** tab.
3. Ensure that **Enable trace** is checked and enter the WebSphere Application Server trace string.

Create missing password scripts

The WebSphere Commerce V5.6 Toolkit lacks support for generation of passwords for WebSphere Commerce and WebSphere Commerce Payments. Follow the instructions in this section to add this support to WebSphere Commerce V5.6 Toolkit.

Create wcs_password script

The WebSphere Commerce V5.6 Toolkit does not ship with a script to generate encrypted user passwords for WebSphere Commerce. Create a file called **wcs_password.bat** in the <wctoolkit_home>\bin directory. Example A-3 shows the contents of the file.

Example: A-3 wcs_password.bat script for WebSphere Commerce V5.6 Toolkit

```
@echo off

%~d0
cd %~p0
setlocal

rem Pick up the WebSphere Commerce environment variables
call setenv.bat
set CP=%RUNTIME_CLASSES%;%PROPERTIES%;%RUNTIME_JARS%\wcs.jar

%JAVA_HOME%\bin\java -classpath %CP% com.ibm.commerce.util.WCSPassword %1 %2 %3

endlocal
```

Create wcs_pmpassword script

The WebSphere Commerce V5.6 Toolkit does not ship with a script to generate encrypted passwords for WebSphere Commerce Payments. Create a file called **wcs_pmpassword.bat** in the <wctoolkit_home>\bin directory. Example A-4 shows the contents of the file.

Example: A-4 wcs_pmpassword.bat script for WebSphere Commerce V5.6 Toolkit

```
@echo off
```

```
%~d0
```

```
cd %~p0
```

```
setlocal
```

```
rem Pick up the WebSphere Commerce environment variables
```

```
call setenv.bat
```

```
set CP=%RUNTIME_CLASSES%;%PROPERTIES%;%RUNTIME_JARS%\wcs.jar
```

```
%JAVA_HOME%\bin\java -classpath %CP% com.ibm.commerce.util.WCSPMPassword %1 %2
```

```
endlocal
```



B

Backup WebSphere Commerce Suite V5.1

This appendix provides instructions for backing up the WebSphere Commerce environment. Two main aspects will be covered, backing up the directory structure where WebSphere Commerce is installed and backup up the database.

This appendix is organized in the following structure:

- ▶ File system
- ▶ Database

File system

The directories where WebSphere Commerce resides along with any other user defined directory needs to be backed up. For example, the default installation directory, configuration files, and any customized files and directories (generally defined for static content).

Default directories

The default directories are made up of the installation directory including all its subdirectories and files. Backing up these directories and files will ensure that all files related to WebSphere Commerce will be included in the backup. In case your file system differs from the default installation directory, be sure that all configuration files for WebSphere Commerce, WebSphere Application Server, and IBM HTTP Server are included in your backup. Since you may need to refer to these directories and files during the migration process, these directories and files should be backed up to a location that is easily accessible.

The following section gives examples of directories and files that should be backed up. These examples are based on the default install directories of WebSphere Commerce Suite V5.1.

- Your instance configuration file:

`C:\WebSphere\WCS\instances\instance_name\instance_name.xml`

where *instance_name* is the name of your instance.

- Your store property files, jsps, images, and other store related data:

`C:\WebSphere\WCS\stores\properties`

`C:\WebSphere\WCS\stores\web`

- The WebSphere Application Server configuration file:

`C:\WebSphere\WAServer\bin\admin.config`

- The IBM HTTP Server configuration file:

`C:\WebSphere\HTTPServer\conf\httpd.conf`

Note: you will also have to backup you SSL certificates files used for the SSL configuration in the `httpd.conf` file

Custom directories

Custom files and directories that have been created for use with WebSphere Commerce should be also backed up. Any components that are associated with WebSphere Commerce such as the database, WebSphere Application Server,

WebSphere Commerce Payments (formerly WebSphere Payment Manager), the Web server, and the IBM Developer Kit Java 2 Technology Edition.

For example, you may have changed the IBM HTTP Server document root to a directory outside the default WebSphere Commerce install directory. This directory would need to be included in the backup. If you have any custom code or jar files located outside of the default WebSphere Commerce install directory, these directories and file would also need to be included in the backup.

Database

The WebSphere Commerce Suite V5.1 database, along with any additional databases that your system requires needs to be backed up. For example, an additional database that would need to be backed up would be the WebSphere Commerce Payments database. Most likely, a database backup process has been implemented for your system. You can use your normal procedures for your site.

Database backup

Based on your configuration, a DB2 database backup can be performed from the database server machine. The following are instructions on how to perform a DB2 backup.

1. Logon to your Windows system as the Windows user ID that created the DB2 database, or that owns the DB2 database.
2. Complete all database transactions by shutting down your WebSphere Commerce Suite V5.1 system.
3. Start a DB2 command window, by typing the following in a command window:

```
db2cmd
```

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 no applications are connected to your database, the following message displays:

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

If any applications are listed as being connected to your database, you will need to locate the source of the application and shut it down. After you have

shut down the connected applications, reissue the db2 list applications command and verify that no data is returned by the status monitor.

5. 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

6. Create a directory where you will back up your databases, for example:

mkdir drive:\db2_backup

7. 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*

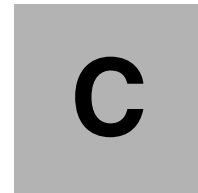
where *db_name* is the name of the database, and *backup_directory* is the full path to where you want to back up the database. Include the drive: letter in the path Draft name. The *backup_directory* must exist. You should receive a message indicating that the back up was successful.

For example, to back up the MALL database to the above backup directory, use the following command:

db2 backup database MALL to drive:\db2_backup

If you have more than one database, repeat the command for each Commerce Suite 5.1 database. You should back up any non-Commerce Suite 5.1 databases, such as the WAS or WAS40 databases, or the WebSphere Commerce Payments (formerly WebSphere Payment Manager) database, PAYMAN, 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*.



Migration scripts

This appendix describes in detail the behavior for each of the scripts we used during the migration.

The appendix is organized in the following sections:

- ▶ WebSphere Commerce V5.6 migration scripts behavior
- ▶ Custom database migration scripts behavior

WebSphere Commerce V5.6 migration scripts behavior

This section discusses the specific execution behaviors of the various migration scripts. This information would be useful for the advanced users who have gone through various test-runs of migration and are concerned about performance tuning the database for quicker migration by understanding the underlying execution behavior of these scripts. If you modify the migration scripts it could potentially result in unsupported migration.

migratedb.bat behavior

This script migrates the database schema and the unencrypted data as needed. The behavior of this script is controlled by one or other configuration file depending on your pre-migration WebSphere Commerce level.

In our case the relevant configuration file would be:

WC56_install_dir/schema/migration/config/DataMigrationPlan51.xml

Here is an excerpt of the configuration file:

Example: C-1 Excerpt of DataMigrationPlan51.xml file

```
<dataMigrationCommand name="drop.wcs.constraint" type="sql">
  <command>
    wcs.drop.referential.sql,
    wcs.drop.view.sql,
    wcs.drop.key.sql,
    wcs.drop.index.sql
  </command>
</dataMigrationCommand>
<dataMigrationCommand name="backup.wcs.bootstrap" type="sql">
  <command>
    wcs.backup.bootstrap.sql,
    wcs.delete.bootstrap.sql,
    wcs.delete.bootstrap_en_US.sql
  </command>
</dataMigrationCommand>
<dataMigrationCommand name="apply.wcs.schema.delta" type="sql">
  <command>
    wcs.fix.null.notnull.sql,
    wcs.schema.new.tables.sql,
    wcs.schema.recreate.tables.sql,
    wcs.schema.alter.tables.sql
  </command>
  <property name="commitCount" value="1" />
</dataMigrationCommand>
```

This file controls the sequence of execution of various SQL files, in addition to other activities. This provides fine grain control over how migration should proceed. Here are some examples of the usefulness of this approach.

Some of the column lengths have decreased in the WebSphere Commerce V5.6 schema when compared to the column lengths in WebSphere Commerce Suite V5.1. However, if you have data that occupies more space than allowed by column length in WebSphere Commerce V5.6 then the data migration from WebSphere Commerce Suite V5.1 schema to WebSphere Commerce V5.6 schema should fail. It is highly recommended that you re-evaluate your need to have larger column length and thus staying with standard and supported WebSphere Commerce V5.6 schema. However, if you must have the same column length as defined in WebSphere Commerce Suite V5.1 schema then you must change the behavior of one of the SQL files mentioned in this configuration file. Both the standard output and the migration log file specifies the command it is currently executing, this corresponds to, say, the `dataMigrationCommand` or `command XML` tag as mentioned in this configuration file. When the data migration discussed above fails you would notice that it was executing the `wcs.schema.recreate.tables.sql` file in the following directory:

```
WC56_install_dir/schema/migration/51
```

You could edit this file to re-apply the same setting as in WebSphere Commerce Suite V5.1.

The other obvious benefit of this execution behavior is that once you know that a given step is taking a long time to execute, you could examine the corresponding SQL statements and tune your database, e.g. if you are running out of bufferpool or log file space then you could pin-point the exact large table of your database causing this behavior and tune your database accordingly.

migrateEncryptedInfo.bat behavior

This script reads all the rows of a given table and then processes and commits a subset of rows at a time. Note that the instance must be migrated prior to running this script as it needs the migrated `instance.xml` file, in addition to certain other files.

Internally, this script calls the `MKChangeApp` command twice with different parameters and config files:

1. The first invocation uses the config file `CCInfoDBUpdate.dbtype.txt`, which defines the following two action items:
 - a. System Key to currentKey migration: Prior to WebSphere Commerce Suite V5.1.1.2 the merchant key was not used to encrypt credit card data,

instead a system key was used. Since this behavior has changed the data must be re-encrypted.

- b. PDI flag enforcement: The variable PDIEncrypt (on | off) in the instance file indicates if credit card data is encrypted. Depending on the value of this variable credit card information is encrypted or decrypted accordingly. This script provides you with an opportunity to change and enforce the PDI encryption setting.
2. The second invocation of the MKChangeApp command uses the config file schema\dbtype\migration\DBUpdate.txt, which defines the following two actions:
 - a. Password migration: Passwords are re-encrypted using the random Salt values and one way hash keys
 - b. Merchant key migration: This script allows you with opportunity to change your merchant key and migrate your data based on the new merchant key.

This script acts on the following tables.

Table 12-1 Tables used by migrateEncryptedInfo script file

TABLE	COLUMNS UPDATED	UNIQUE ID	Notes
USERREG	LOGONPASSWORD, SALT	USERS_ID	Password related information
ORDERPAYINFO	VALUE	ORDERPAYINFO_ID	Credit Card related information
ORDPAYMTHD	PAYDEVICE	ORDERS_ID, PAYMETHOD, PAYDEVICE, REFUNDNUMBER	Credit card related information
PATTRVALUE	STRINGVALUE ENCRYPTFLAG	PATTRVALUE_ID	Personalization attributes, handled with credit card data processing (table began to exist in v5.4 onwards)

wcim.bat behavior

Here is a brief description on what WCIM does at the high level.

The following two are additional steps for Switch-Over instance migration (for Runtime environment only). You will not need to do these before migrating a local WebSphere Commerce Suite V5.1 instance or a WebSphere Commerce Studio V5.1 instance.

1. Package WCIM on v5.6 machine and its environment for remote execution: Run WCIM to create a zip file including WCIM itself as well as the environment it requires. This package will be used to build the migration environment for the Switch-Over migration.
2. Unpack WCIM on v5.1 machine: Unzip the WCIM zip package generated above. This builds the runtime environment with correct level of JDK, XML parser, ANT, Configure Manager client, WCIM while maintaining the file permission and ownership. Note that since the package created includes all these platform dependent files, scripts and attributes you can not migrate across platforms.

Migrating a local WebSphere Commerce Suite V5.1 instance or a WebSphere Commerce Studio V5.1 instance:

1. Backup the instance to be migrated: Backup of the WebSphere Commerce runtime environment is required in case of Switch-Over instance migration. It is optional, but recommended, in other instances. The backup includes:
 - Configuration and installation information of the previous WebSphere Commerce install
 - The configuration information of the WebSphere Commerce instance to be migrated
 - The configuration information of the local IBM HTTP Server
 - The configuration information of WebSphere Commerce Payments and supported payment cassettes and the deployed EAR.
 - If there is no EAR, WCIM will collect enough information to be used to build a migrated EAR. All files are compressed into a single ZIP backup-instance.zip to be transferred to the target WebSphere Commerce server.
2. Migrate the instance: This includes:
 - Creating a migrated EAR based on the original EAR and the default EAR in the new release.
 - Stores published before migration are wrapped into a Web module. Multiple Web modules and resource adaptors are added.
 - EJBs for the correct database type, JSPs, servlets, commands, HTML files, images from the new version are packaged into the migrated EAR, too.
 - An application server 'WC_xxxx' is created with the correct configuration required to run the migrated EAR.

- The JDBC provider and the datasource is configured for the server. The migrated EAR is installed onto the server.
 - The deployment descriptor is modified as well.
 - Configuration of the original instance, including instance itself, access control, organization, trading, loader, payment, and store JSPs are migrated.
 - The Web server host name and the database name / SID can also be changed by WCIM during the migration if users choose to do so.
 - For the Switch-Over migration, the correct Web server host name, WebSphere Application Server node name and WebSphere Application Server cell name are configured for the migrated instance.
 - For local replace migration, the same Web server host name is assumed but user can take this opportunity to switch to a different Web server. For local coexistence migration, a different Web server host name is enforced. Local IHS is configured. WebSphere Application Server plug-in configuration is regenerated. Backup step and migration step can be executed together for local migration. No human intervention is required in between.
 - For local coexistence migration or 5.5 to 5.6 local replace migration, a temporary host name is required for migration. WCIM can perform IP-switching to reconfigure the migrated instance to use the final Web server host name. WebSphere Application Server virtual hosts, instance configuration XML file and local IHS configuration are modified. WebSphere Application Server plug-in configuration is regenerated.
3. The development environment is migrated by WCIM in a similar fashion as detailed in the steps 3 and 4. In addition, custom store assets, servlets, commands are moved to pre-configured projects of Stores and WebSphereCommerceServerExtensionLogic in the new workspace.

Note: The WCIM tool will convert all occurrences of `<%@ page language="JAVA" %>` to `<%@ page language="java" %>` during this step, although we found that not all occurrences was modified. After this step ensure that no more occurrences of `<%@ page language="JAVA" %>` exists. This can be done by doing a file content search

Custom database migration scripts behavior

To ease the task of migrating the database we created custom scripts which would run prepare, migrate and post migration tasks on the database. In this section we describe the behavior of the scripts.

The batch scripts consists of:

► testrun.bat

This file is the initial start file. This will call migration.bat with the correct parameters.

► migration.bat

This file will perform the actual migration. It calls other scripts in the following order:

i. pre.migration.bat

ii. migratedb.bat (WebSphere Commerce V5.6 script)

iii. post.migration.bat

► pre.migration.bat

This file will perform all the pre-migration tasks.

► post.migration.bat

This file will perform the post-migration tasks

The sequence in when the the scripts are executed is described in Figure 12-1 on page 261

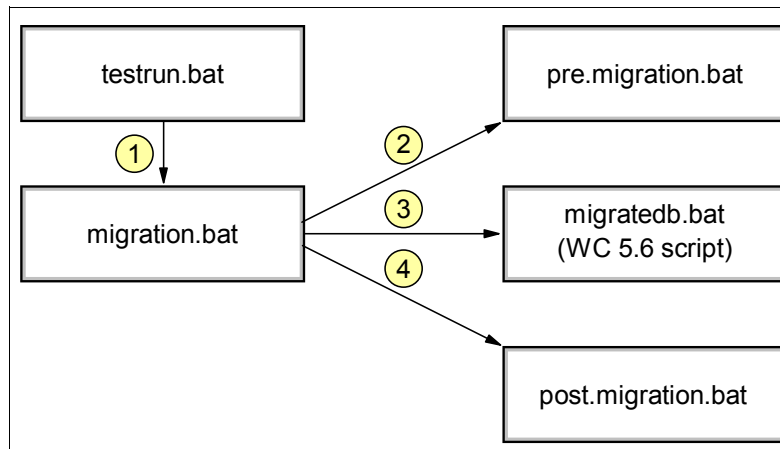


Figure 12-1 Sequence diagram of the batch files

Database preparation batch script

The script **pre.migration.bat** prepares the database for migration. This script takes the following parameters:

1. Database name

2. DB user id
3. DB password
4. Log directory
5. Instance name

The script has the following behavior:

1. Set parameters as local variables
2. Call updateDB2Configuration.bat to update the database attributes
3. Update the database attribute logfilesize to 30000
4. Stop and start DB2 for changes to take effect
5. Connect to database
6. Execute **pre.migration.sql** file, refer to, “Database preparation SQL script” on page 263 for more details
7. Close connection

The content of the pre.migration.bat file:

Example: C-2 Content of the pre.migration.bat file

```
@echo off
```

```
set database=%1
set user=%2
set passwd=%3
set logdir=%4
set instance=%5
set logfile=%logdir%\pre.migration.log
```

```
echo Update DB2 attributes
```

```
setlocal
```

```
call %WCS_HOME%\bin\updateDB2Configuration.bat %database% %logdir%\updatedb2configuration.log
endlocal
```

```
db2 update db configuration for %database% using logfilsiz 30000
```

```
echo Stop and restart DB2.
```

```
db2stop
```

```
db2start
```

```
echo Connect to database
```

```
db2 connect to %database% user %user% using %passwd%
```

```
echo Call pre migration script
```

```
db2 -tvf pre.migration.sql > %logfile%
```

```
echo Disconnect
db2 terminate
```

Database preparation SQL script

The SQL file executes the changes needed on the database needed to prepare the database for migration.

The **pre.migration.sql** file will perform the following tasks:

1. Clear MSGSTORE table (not to be used on production database, refer to Section 10.3, “Database preparation” on page 188 for more information)
2. Update status on ORDERS and ORDERITEMS table
3. Update administrators profile type
4. Remove roles for our administrators and add appropriate roles
5. Create master catalog
6. Update KEYS table for the targeted tables

The content of the pre.migration.sql file:

Example: C-3 Content of the pre.migration.sql file

```
-- CLEAR MSGSTORE TABLE
-- NOTE : MESSAGES IN PRODUCTION ARE DEALT WITH DIFFERENTLY
delete from MSGSTORE;

-- UPDATE STATUS FROM C TO S
update orders set status='S' where status='C';
update orderitems set status='S' where status='C';

-- UPDATE ADMINISTRATORS (REGISTERTYPE IN ('A','S'))
-- AND THAT HAVE ENTRIES IN ACCMBRGRP TO HAVE PROFILETYPE='B'
update users set profiletype='B' where registertype in ('A','S') and (select count(*) from
accmbrgrp where accmbrgrp.users_id=users.users_id)>1;

-- USERS ADMIN#1 AND ADMIN#2 ARE USED BY THE CUSTOMER FOR ACCELERATOR
-- THEY HAVE ROLE -3=CUSTOMER SERVICE REPRESENTATIVE AND -5=ORDERCLERK AND -8=MERCHANDISING
MANAGER
-- REMOVE ALL THESE ROLES AND ADD -6=STORE ADMINISTRATOR IN THE POST MIGRATION
delete from accmbrgrp where users_id in (<users_id>);

-- ADD STORE ADMINISTRATOR ROLE TO THEM
insert into accmbrgrp values (-6,<users_id>,-2000,NULL);
insert into accmbrgrp values (-6,<users_id>,-2000,NULL);
```

```
-- CREATE NEW MASTER CATALOGUE
insert into catalog values (10002,-2000,'Master Catalog','Master Catalog');
insert into catalogdsc values (10002,-1,'Master Catalog',NULL,NULL,NULL,NULL);

insert into catgroup values ((select max(catgroup_id) from catgroup)+1,-2000,'Master
Category',0,NULL,NULL,NULL,NULL);
insert into catgrpdesc (catgroup_id,language_id,name,published,shortdescription) values((select
max(catgroup_id) from catgroup),-1,'Master Category',1,'Master Category description');

insert into cattogrp values (10002,(select max(catgroup_id) from catgroup));

insert into storecat values (10001,10001);
insert into storecat values (10002,10001);
insert into storecgrp values (10001,(select max(catgroup_id) from catgroup));

-- ADD ALL CATENTRIES INTO ABOVE CATGROUP
insert into catpenrel(catgroup_id, catalog_id, catentry_id, rule, sequence) select (select
max(catgroup_id) from catgroup), 10002, catentry_id, '-', 0.0 from catentry where catentry_id
<> 0;

-- UPDATE KEYS FOR CATALOG,CATGROUP
update keys set counter=(select max(counter) from keys where tablename='catalog')+1 where
tablename='catalog';
update keys set counter=(select max(counter) from keys where tablename='catgroup')+1 where
tablename='catgroup';
```

Database migration batch script

The script **migration.bat** migrated the database. This script takes the following parameters:

1. Database name
2. DB user id
3. DB password
4. Log directory
5. Instance name

The script has the following behavior:

1. Set parameters as local variables
2. Call WebSphere Commerce V5.6s **config_env.bat** to set environment variables
3. Call **pre.migration.bat** to prepare the database

4. Call WebSphere Commerce V5.6 database migration script **migratedb.bat** to migrate the database
5. Call **post.migration.bat** to perform the final post tasks

The content of the migration.bat file:

Example: C-4 Content of the migration.bat file

```
@echo off

set database=%1
set user=%2
set passwd=%3
set logdir=%4
set instance=%5

echo Set environment
call config_env.bat

echo Call pre migration script
call pre.migration.bat %database% %user% %passwd% %logdir% %instance%

echo Call commerce migration script (use -precheck option until no errors are reported)
call %wc56installldir%\bin\migratedb -dbtype db2 -dbname %database% -dbuser %user% -dbpass
%passwd% -from 51 -instance %instance%

echo Call post migration script
call post.migration.bat %database% %user% %passwd% %logdir% %instance%
```

Database post migration batch script

The script **post.migration.bat** performs the final tasks on the database to finish the migration. This script takes the following parameters:

1. Database name
2. DB user id
3. DB password
4. Log directory
5. Instance name

The script has the following behavior:

1. Connect to the database
2. Execute **post.migration.sql** file, refer to, “Database post migration SQL script” on page 266 for more details

3. Apply patch for **migrateencryptedinfo.bat** into **Utilities.jar**, refer to Chapter 11, “Migrating WebSphere Commerce components” on page 201 for more details on why this patch is applied
4. Call WebSphere Commerce V5.6 encrypted data migration script **migrateencryptedinfo.bat** to migrate the encrypted data
5. Execute **choosemc.sql** to assign the master catalog

Database post migration SQL script

The SQL file executes the final changes needed on the database to finish the database migration.

The **post.migration.sql** file will perform the following tasks:

1. Correct the index errors introduced by **migratedb.bat**
2. Remove records with non-encrypted values starting with space. Refer to Chapter 11, “Migrating WebSphere Commerce components” on page 201 for more details on this.

The content of the post.migration.sql file:

Example: C-5 Content of post.migration.sql file

```
-- CORRECT INDEX ERRORS DURING DB MIGRATION
create index I0000511 on campaign ( storeent_id asc);

drop index I0000539;
create index I0000539 on contract ( member_id asc);

create index I0000588 on initiative ( storeent_id asc);

drop index I0000824;
create index I0000824 on userpvdev ( users_id asc);

-- DELETE ORDPAYINFO VALUES THAT STARTS WITH ' ' SO IT IS NOT ASSUMED TO BE A ENCRYPTED VALUE
delete from ordpayinfo where value like ' %';
```



Moving from single to multi-node environment

This appendix describes the steps needed to move from a single-node runtime implementation to a multi-node remote database runtime environment after the migration took place.

Most of the steps described in this appendix are based on what we have already documented in previous chapters.

This appendix is organized in the following sections:

- ▶ Single-node environment
- ▶ Installing and configuring the database node
- ▶ Moving the database
- ▶ Connecting to the remote database

Single-node environment

In Part 3, “Production environment” on page 161 we discussed how to install WebSphere Commerce V5.6 on a single-node runtime environment and create a test instance to verify the installation. Later we migrated an entire application from WebSphere Commerce Suite V5.1 to WebSphere Commerce V5.6.

Having a running single-node runtime environment is the starting point for moving to a multi-node, remote database runtime environment.

The following sections provides a high level overview of the steps that need to be carried out to install and configure a multi-node remote database runtime environment.

Installing and configuring the database node

In previous chapters we described how to install DB2 Universal Database V8.1 for the development environment. Most of those steps can be repeated to install the database on a remote node.

Create a Windows user for DB2

Create a Windows user with the same user_id, password and user rights that the one used in the single-node environment. To create this user follow these steps:

1. Click **Start -> Settings -> Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Computer Management**.
4. In the tree on left of the new window, expand **Local Users and Groups** and click **Users**.
5. Right-click in the right-hand pane and select **New User...** from the menu.
6. Enter the user name used for your original production database.
7. Enter the corresponding password in the Password and Confirm Password fields.
8. Deselect **User must change password at next logon**.
9. Select **User cannot change password** and **Password never expires**.
10. Click **Create**.
11. Click **Close**.

12. Still in the Computer Management window, right-click on the user you just created and select **Properties** from the menu.
13. Click the **Member Of** tab and click the **Add...** button.
14. Select **Administrators** from the list of user groups and click **Add**.
15. Click **OK** to add the user to the group and then click **OK** to close the user properties window.
16. Close the Computer Management window.

Installing DB2

To install DB2 Universal Database V8.1, complete the following steps:

1. Run **setup.exe** from the root of the DB2 installation CD.
2. When the Launchpad windows opens, click **Install Products**.
3. Click **Next** to start the installation wizard.
4. On the welcome screen, click **Next**.
5. Select the option to accept the terms and conditions and click **Next**.
6. Select the **Typical** option (default) and click **Next**.
7. When the warning screen appears, click **OK** to continue.
8. Ensure that **Install DB2 Enterprise Server Edition on this computer** is selected and click **Next**.
9. Select the path to install DB2 Universal Database V8.1 and click **Next**.
10. In the User Information frame, enter the user name and password of the user that you created in "Create a Windows user for DB2" on page 268. Ensure that the **Use the same user name and password for the remaining DB2 services** check box is checked and click **Next**.
11. In the Administration contact list location frame, select **Local** and click **Next**.
A warning window will appear to tell you that maintenance notifications will not be sent by DB2. Click **OK** to continue.
12. On the Configure DB2 instances screen, click **Next** to accept the defaults.
13. Unless you need to use the Task Center or the DB2 scheduler, ensure **Do not prepare the DB2 tools catalog on this computer** is selected and click **Next**.
14. On the Specify a contact for health monitor notification screen, select **Defer the task until after installation is complete** and click **Next**.
15. Review the settings in the text box and click **Install** to begin the installation.
16. When the window appears to tell you that Setup is complete, click the **Finish** button.

If you are prompted to restart your computer, do so at this point.

After a few seconds, the *First Steps* window will appear. Close this window.

Moving the database

In order to move the database to the remote database node it is necessary to backup, transfer and restore the current database into the remote node.

These steps are summarized as follows:

- ▶ take the production WebSphere Commerce node offline. Ensure the commerce engine is stop and there are no active connections to the commerce database.
 - stop the commerce server
`<WAS_home>\bin\stopServer <WC_server_name>`
 - stop all connections to the database
`DB2 terminate`
`DB2 force application all`
- ▶ backup the current production database
`DB2 backup db <WC_db> to <backup_dir>`
- ▶ transfer the backup directory to the remote database node
- ▶ restore the database from the backup directory
`DB2 restore db <WC_db> from <backup_dir>`

Connecting to the remote database

In order to access the remote database you just restored first you have to drop the actual production database from the WebSphere Commerce node and then create a connection and an alias for the remote database node.

These steps are summarized as follows:

- ▶ on the WebSphere Commerce node drop the commerce database
`DB2 drop db <WC_db>`
- ▶ create a connection from the WebSphere Commerce node (DB2 client) to the remote TCP/IP node
`DB2 catalog tcpip node <node_name> remote <remote_db_hostname> server <server_port>`

- create a local alias for the remote database

```
DB2 catalog db <WC_db> at node <node_name>
```

Verify connectivity

To verify the connectivity with the remote database node from the WebSphere Commerce node type the following command:

```
DB2 connect to <WC_db> using <db_user_id> using <db_user_password>
```

You should receive the Database Connection Information display with the database server version, user ID used in the connection and the database name.

For further information about configuring remote connections on DB2 Universal Database refer to the product Information Center at the following URL:

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>



Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

<ftp://www.redbooks.ibm.com/redbooks/SG246320>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG24-6320.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<i>File name</i>	<i>Description</i>
DB2CMS.zip	Zipped file containing the DB2 custom migration scripts.

sample.zip

Zipped containing the rmPaymentsServer.bat script and a copy of the product.xml.51.sample file.

How to use the Web material

Create a subdirectory on your workstation, and unzip the contents of the Web material zip file into this folder. Detailed instructions on how to use these scripts are provided in Chapter 9, “Installing WebSphere Commerce V5.6” on page 163 and in Appendix C, “Migration scripts” on page 255

Abbreviations and acronyms

This abbreviations and acronyms file is optional.

Use this file by adding names and descriptions to it. Sort these names: **highlight rows >Table > Sort > Sort By: Column 1 > Sort** or optionally add names and descriptions to the Index file instead of this file by indexing the first use of an abbreviation or acronym: **highlight text > Special > Marker > Index > New Marker**

abbreviation1	Description1
abbreviation2	Description2
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
abbreviation3	Description3
abbreviation4	Description4

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 278. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *WebSphere Commerce V5.5 Handbook, Customization and Deployment Guide - SG24-6969-00*
- ▶ *WebSphere Commerce Suite V5.1 Handbook - SG24-6167*

Other publications

These publications are also relevant as further information sources:

- ▶ *Migration Guide for Windows 2000 and Windows 2003 - migrating from WebSphere Commerce Suite V5.1*
- ▶ *Migration Guide for WebSphere Commerce Developer - migrating from WebSphere Commerce Studio V5.1*

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ WebSphere Commerce information center
<http://publib.boulder.ibm.com/infocenter/wc56help/index.jsp>
- ▶ WebSphere Application Server technote 21157884: Global Security Kit (GSKit) install fails whe installing fixpack 2 on V5.0
<http://www.ibm.com/support/docview.wss?uid=swg21157884>
- ▶ Microsoft Windows Update
<http://www.windowsupdate.com>
- ▶ Enterprise JavaBeans specification

<http://java.sun.com/products/ejb/docs.html>

- ▶ J2EE Connector architecture

<http://java.sun.com/j2ee/connector/>

- ▶ WebSphere Payment Manager support

<http://www.ibm.com/software/genservers/commerce/paymentmanager/support.html>

- ▶ DB2 Information Center

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

- access bean
 - ibm-ejb-access-bean.xml 145
 - string converters 144
- Access control 17
 - policies 30
 - roles 30
 - Subscription based model 30
- access isolation level 146
- Accounts 29
- Ad Copy 31
- adapters
 - CrossWorlds 28
- administration
 - rule server 155
- Administration Console 22
 - trace strings 244
- Administrative processes
 - business processes 23
- Analytics 32
 - Authorization 32
 - Data Mart 32
 - Reporting Framework 32
 - Single logon 32
 - Tivoli Web Site Analyzer 32
- application.xml 128
- Architecture
 - WebSphere Commerce components 10
 - WebSphere Commerce subsystems 15
- Authentication 17

B

- B2B direct 24
- backup
 - custom directories 252
 - database 253
 - directories 252
- BankServACH 27
- behavior
 - migration scripts 256
- Blaze Advisor 31
- business model 23
 - B2B direct 24

- Consumer direct 23
- demand chain 24
- Direct sales 23
- hosting 25
- Hosting Ex-Sites 25
- supply chain 25
- Value chain 24
- business processes
 - Administrative processes 23
 - Solution 23
 - Starter stores 23

C

- cache 38
- Caching 33
- calculation framework 132
- Campaigns 31
- Cassettes
 - BankServACH 27
 - CustomOffline 27
 - OfflineCard 27
 - Paymentech 27
 - VisaNet 27
- catalog
 - master catalog 213
- Catalog entries 19
 - classification 28
- Catalog filtering 28
- Catalog import 28
- Catalog Management 28
- Catalog Manager 27
- Catalog subsystem 19
- class
 - custom 135
- codes
 - SQL0601N 212
 - SQL0605W 212
- Collaboration 14, 30
- Collaborative Workspaces 14
- commands 133
 - depolyment 233
 - framework 131
 - logon 132

- tasks 153
- Commerce analytics 13
- Configuration Manager 21, 27
 - start 242
- considerations
 - planning 40
- Consumer direct 23
- container transaction 149
- Cookie 18
- Coupons 31
- create password 249
- credit cards 214
- CrossWorlds 28
- custom class 135
- custom code 39, 129
- custom directories
 - backup 252
- custom interface 135
- Customer Care 14
- CustomOffline 27

D

- data bean 153
- database
 - backup 253
- databeans
 - deployment 233
- db2 backup 254
- demand chain 24
- Deploying commands 233
- Deploying databeans 233
- Deploying EJB
 - configuring JAR file 231
 - exporting JAR file 231
 - importing JAR file 231
- Deploying store assets 234
- deployment.xml 127
- Developer 26
- development environment 3, 68
- Direct Sales
 - B2B direct 24
 - Consumer direct 23
- Direct sales 23
- directories
 - backup 252
- discount models 31
- downtime 46
- DynaCache 38

E

- EJB
 - deployment 231
- ejbCreate 149, 151
- ejbPostCreate 151
- encrypt password 249
- encrypted data
 - credit cards 214
 - ORDPAYINFO 214
 - ORDPAYMTHD 214
 - PATTRVALUE 214
 - USERREG 214
 - users 214
- environment
 - development 3
 - production 3
- Extended Sites 25

F

- FinderHelper 151
- framework
 - calculation 132
 - WC commands 131
- functionality 38

G

- Globalization support 19
- Groupings 19

H

- hosting 25
- Hosting Ex-Sites 25
- Hosting Extended Sites
 - hosting 25
- HTML content 230
- httpd.conf 230

I

- IBM Directory Server 14
- IBM HTTP Server
 - start 244
 - stop 244
- ibm-ejb-access-bean.xmi 145
- ibm-ejb-jar-ext.xmi 152
- IHS configuration 230
- in-place migration approach 46
- in-place migration scenario 48

- instance
 - start 240
 - status 240
 - stop 241
- InterChange Server 28
- interface
 - custom 135
- interfaces
 - FinderHelper 151
- inventory management 31
- Inventory subsystem 20
- isolation level
 - change access 146

J

- J2EE Connector Architecture 152
- java.rmi.RemoteException 150
- JCA 152

K

- key
 - merchant 214
 - SALT 214

L

- LDAP 14, 16, 19
- Loader Package 27
- logon command 132
- looseconfig.xmi 126
- Lotus QuickPlace 14
- Lotus Sametime 14

M

- managing
 - Configuration Manager 242
 - IBM HTTP Server 244
 - WebSphere Commerce 240
 - WebSphere Commerce Payments 241
- mapping
 - WC commands 133
- Marketing subsystem 20
- Master Catalog 28
- master catalog 213
- Member groups 15
- Member security services 17
- Member subsystem 15
 - security services 17

- single sign-on 19
- user registration 15
- Members 15
- Merchandising associations 19
- Merchandising subsystem 20
- merchant key 214
- Messaging integration 13
- Messaging subsystem 21
- methods
 - data bean 153
 - ejbCreate 151
 - ejbPostCreate 151
 - java.rmi.RemoteException 150
 - WCSecurity 148
- migratedb.bat 256
- migrateEncryptedInfo.bat 257
- migrating
 - IHS configuration 230
- migrating static content 230
- migration approach
 - in-place 46
 - Switch-Over 46
- migration scenario
 - in-place 48
- migration scripts 256
 - migratedb.bat 256
 - migrateEncryptedInfo.bat 257
 - migration.bat 264
 - post.migration.bat 265
 - post.migration.sql 266
 - pre.migration.bat 261
 - pre.migration.sql 263
 - WebSphere Commerce Instance Migration (WCIM) 202
- migration strategy 38
 - custom code 39
 - DynaCache 38
 - functionality 38
 - optimization 38
 - product management 39
 - user management 39
- migration.bat 264
- MQ 16
- myAccount.jsp 156

N

- NAT 46
- Navigational Catalog 28

Network Address Translation 46

O

OfflineCard 27
 Online registration 16
 optimization 38
 order management 31
 Order subsystem 20
 ORDPAYINFO 214
 ORDPAYMTHD 214
 Organization Administration Console 22
 Organizational entity 15
 Organizational unit 15
 overview
 development environment 68
 product 8
 production environment 184

P

package file
 WCIM 204
 password
 create 249
 encrypt 249
 Password Manager 22, 27
 password reset
 webadmin 244
 PATTRVALUE 214
 Paymentech 27
 Personalization 13
 planning considerations 40
 plugin-cfg.xml 11
 post.migration.bat 265
 post.migration.sql 266
 pre.migration.bat 261
 pre.migration.sql 263
 price calculations 153
 pricing 153
 product
 external management 39
 Product Advisor 35, 154
 Product Management 28
 Product Management tooling 30
 product overview 8
 production environment 3
 overview 184
 Promotions 31

R

Redbooks Web site 278
 Contact us xvi
 register.jsp 156
 registration
 users 131
 requirements
 Development
 hardware 43
 software 43
 Production
 hardware 41
 software 42
 skills 40
 reset password
 webadmin 244
 Rule server
 administration 155

S

SALT key 214
 sample stores 23
 scripts
 migration 256
 SQL 213
 scripts behavior
 migratedb.bat 256
 migrateEncryptedInfo.bat 257
 migration.bat 264
 post.migration.bat 265
 post.migration.sql 266
 pre.migration.bat 261
 pre.migration.sql 263
 wcs_password.bat 249
 wcs_pmpassword.bat 249
 security 32
 serialVersionUID 149
 server
 downtime 46
 serverStatus 240
 service
 rules 155
 Session control 18
 Session Manager 18
 Session types 18
 settings
 trace 248
 Single sign-on 19

- SKU generation 29
- SOAP 245
- Solution
 - business processes 23
- SQL codes
 - SQL0601N 212
 - SQL0605W 212
- SQL scripts 213
- Starter stores
 - business processes 23
- startServer 240
- static content 230
- store assets
 - deployment 234
- strategy 38
- string converters 144
- Subscription based model 30
- Subsystems 15
 - catalog 19
 - inventory 20
 - marketing 20
 - members 15
 - merchandising 20
 - orders 20
 - trading 20
- supply chain 25
- Switch-Over migration scenario 46

T

- table
 - ORDPAYINFO 214
 - ORDPAYMTHD 214
 - PATTRVALUE 214
 - USERREG 214
- task commands 153
- Test Environment
 - full 248
 - lightweight 247
- toolkit 68
- tools
 - accelerator 21
 - administration console 22
 - configuration manager 21
 - organization administration console 22
 - password manager 22
- trace settings 248
- trace strings 244
- Trading subsystem 20

U

- URL rewriting 18
- user
 - external management 39
- user registration 131
- User registration methods 15
 - LDAP 16
 - Loader package 17
 - MQ 16
 - online 16
- USERREG 214
- users 15, 214

V

- Value chain 24
 - demand chain 24
 - supply chain 25
- VisaNet 27
- VisualAge for Java 26

W

- WC commands 133
- WCIM
 - backupwc_51_.zip 205
 - backupwcim.zip 204
 - instance_backup.zip 205
 - migration scripts 202
 - package file 204
- wcs_password.bat 249
- wcs_pmpassword.bat 249
- WCSecurity 148
- webadmin
 - password reset 244
- WebSphere Business Portals 14
- WebSphere Commerce 240
 - administration tools 9
 - business interaction engine 9
 - common server runtime 8
 - software components 8
 - Database Server 12
 - enablement software 13
 - Web Server 11
 - WebSphere Application Server 11
 - WebSphere Commerce Payments Server 13
 - WebSphere Commerce Server 12
 - subsystems 8, 15
 - Catalog subsystem 19

Inventory subsystem	20	Product Management tooling	30
marketing subsystem	20	Promotions	31
Member subsystem	15	security	32
Merchandising subsystem	20	SKU generation	29
Messaging subsystem	21	WebSphere Commerce Payments	27
Order subsystem	20		
Trading subsystem	20		
WebSphere Commerce Payments	27, 241		
WebSphere Commerce Server			
components			
subsystems	15		
WebSphere Commerce Tools	21		
Accelerator	21		
Administration Console	22		
Configuration Manager	21		
Organization Administration Console	22		
Password Manager	22		
WebSphere plug-in	11		
WebSphere Test Environment	26		
Full	248		
Lightweight	247		
What's New	26		
Access Control	30		
Accounts	29		
Ad Copy	31		
Analytics	32		
Business Models	30		
Caching	33		
Campaigns	31		
Catalog entries	28		
Catalog filtering	28		
Catalog import	28		
Catalog Management	28		
Catalog Manager	27		
Collaboration	30		
Configuration Manager	27		
Coupons	31		
CrossWorlds	28		
Developer	26		
discount models	31		
InterChange Server	28		
inventory management	31		
Loader Package	27		
Master Catalog	28		
Navigational Catalog	28		
order management	31		
Password Manager	27		
Product Advisor	35		
Product Management	28		

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(->Hide:-)Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.frm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

6320spine.frm 285



Keeping Commerce Applications Updated

WebSphere Commerce 5.1 to

(1.5" spine)
1.5"<-> 1.998"
789 <-> 1051 pages



Keeping Commerce Applications Updated

(1.0" spine)
0.875"<-> 1.498"
460 <-> 788 pages



Keeping Commerce Applications Updated WebSphere

(0.5" spine)
0.475"<-> 0.875"
250 <-> 459 pages



Keeping Commerce Applications Updated WebSphere Commerce 5.1

(0.2" spine)
0.17"<-> 0.473"
90 <-> 249 pages

(0.1" spine)
0.1"<-> 0.169"
53 <-> 89 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.frm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.
Draft Document for Review July 28, 2004 7:33 pm

6320spine.frm 286



Redbooks

Keeping Commerce Applications Updated WebSphere Commerce

(2.5" spine)
2.5"<->nnn.n"
1315<-> nnn pages



Redbooks

Keeping Commerce Applications Updated WebSphere Commerce

(2.0" spine)
2.0"<-> 2.498"
1052 <-> 1314 pages



Draft Document for Review July 28, 2004 7:34 pm

Keeping Commerce Applications Updated WebSphere Commerce



Migration Strategy and Planning

This IBM Redbook will assist you in the migration of WebSphere Commerce V5.1 applications. The migration path includes the development and production environments. This guide presents a migration strategy and planning, migration tools, and practical migration examples.

Production and Development Environments

The development environment migration topics include migration from VisualAge for Java V3.5.2 and WebSphere Commerce Studio V5.1 to WebSphere Studio Application Developer V5.1.1 included in WebSphere Commerce Developer V5.6. Step-by-step instructions for many practical migration examples are provided.

Step-by-Step Instructions

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks