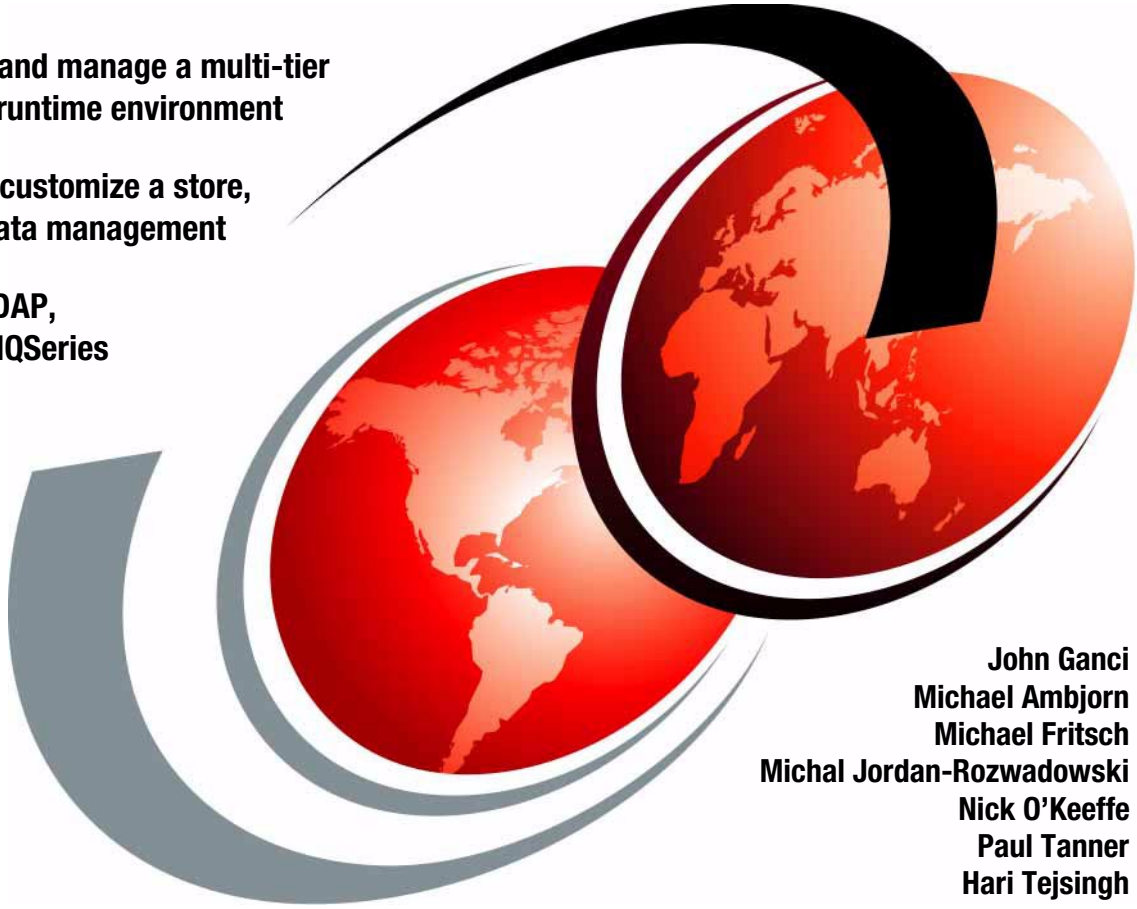
IBM

WebSphere Commerce Suite V5.1 Handbook

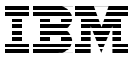
- Implement and manage a multi-tier commerce runtime environment
- Create and customize a store, auctions, data management
- Integrate LDAP, Payment, MQSeries



John Ganci
Michael Ambjorn
Michael Fritsch
Michal Jordan-Rozwadowski
Nick O'Keeffe
Paul Tanner
Hari Tejsingh

Redbooks

ibm.com/redbooks



International Technical Support Organization

WebSphere Commerce Suite V5.1 Handbook

August 2001

SG24-6167-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 615.

First Edition (August 2001)

This edition applies to WebSphere Commerce Suite V5.1, for use with Windows NT, Windows 2000, AIX, and Solaris.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195

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

© Copyright International Business Machines Corporation 2001. All rights reserved.
Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface xvii
The team that wrote this redbook xviii
Special notice xx
IBM Trademarks xx
Comments welcome xx

Part 1. Introduction to WebSphere Commerce Suite V5.1 1

Chapter 1. Introduction 3
1.1 Overview 4
1.1.1 e-commerce models 5
1.1.2 Concepts of e-commerce 5
1.2 Functionality 6
1.2.1 Business enhancements 6
1.2.2 Technical enhancements 8
1.2.3 Management and administration tools 11
1.3 Product packaging 12
1.3.1 Runtime product packaging 12
1.3.2 Development product packaging 12

Chapter 2. Skills planning 15
2.1 Roles and skills 16
2.1.1 I/T specialist roles 16
2.1.2 Business user roles 18
2.2 Matching skills to customization requirements 19
2.2.1 Basic customization 19
2.2.2 Intermediate customization 20
2.2.3 Advanced customization 20
2.3 Education 21
2.3.1 WebSphere Commerce Suite V5.1 courses 21
2.3.2 WebSphere and VisualAge for Java courses 22
2.3.3 Education roadmaps 22
2.4 Where to find more information 23
2.4.1 Product documentation and online help 24
2.4.2 IBM Redbooks 25
2.4.3 IBM Learning Services 26
2.4.4 IBM certifications 26

Chapter 3. Runtime architecture 29

© Copyright IBM Corp. 2001

3.1 Business considerations 30
3.1.1 Scalability 30
3.1.2 Availability 30
3.1.3 Security 31
3.1.4 Back-end integration 31
3.1.5 Skills 31
3.1.6 Cost 31
3.2 Runtime configurations 32
3.2.1 Single-tier runtime configuration 32
3.2.2 Two-tier runtime configuration 33
3.2.3 Three-tier configuration 35
3.2.4 Enterprise 3-tier configuration 36
3.3 Runtime architecture components 37
3.3.1 Software components 38
3.3.2 Commerce Suite runtime architecture components 39
3.4 Runtime computing flow 41
3.5 Where to find more information 42

Chapter 4. Programming model 43
4.1 Concepts 44
4.1.1 Java 2 Platform, Enterprise Edition (J2EE) 44
4.1.2 Model-view-controller (MVC) design pattern 46
4.2 Application development overview 47
4.2.1 Application architecture 47
4.2.2 Approaches to development 48
4.3 Web Controller 50
4.3.1 URL registry 52
4.3.2 Command registry 53
4.3.3 View registry 53
4.4 Commands 54
4.4.1 Controller commands 54
4.4.2 Command factory 56
4.4.3 Task commands 56
4.5 Data beans 57
4.5.1 Data bean manager 57
4.5.2 Data bean implementation 57
4.6 Views 57
4.6.1 View commands 57
4.6.2 JavaServer Pages (JSPs) 58
4.7 Error handling and messages 58
4.7.1 Exception handling framework 58
4.7.2 Customizing error handling and messages 59
4.8 Debugging methods 60

4.8.1 WebSphere Test Environment (WTE) 60
4.8.2 WebSphere Commerce Suite log 61
4.8.3 Program trace messages 61

Part 2. Implementing the runtime environment 63

Chapter 5. WCS runtime environment implementation overview 65
5.1 Installation methodologies 66
5.2 High-level installation steps: 1, 2, 3-tier 67
5.2.1 High-level installation steps: 1-tier 68
5.2.2 High-level installation steps: 2-tier 70
5.2.3 High-level installation steps: 3-tier 73
5.3 WCS instance 76
5.3.1 WCS instance overview 77
5.3.2 Configuration Manager: WCS instance creation 80
5.3.3 Command line: WCS instance creation 80
5.3.4 Manual database creation - Loader package 80
5.3.5 Multiple instances 84
5.3.6 Deleting a WCS instance 84
5.4 Where to find more information 85

Chapter 6. WCS runtime for Windows NT and 2000: DB2 and IHS 87
6.1 WCS runtime environment for Windows 88
6.1.1 Overview 88
6.1.2 Hardware and software prerequisites 89
6.1.3 Hardware used in our test environment 89
6.1.4 Software used in our test environment 90
6.1.5 Install Windows NT/2000 and service packs 90
6.2 Install the IBM HTTP Server 91
6.2.1 Install HTTP Server 91
6.2.2 Configure the HTTP Server 92
6.2.3 Verify the HTTP Server 95
6.3 Install the DB2 Server 96
6.3.1 Install the DB2 V7.1 Server 96
6.3.2 Verify the DB2 V7.1 Server installation 98
6.3.3 Install the DB2 V7.1 Fixpack 2a 98
6.3.4 DB2 V7.1 Server configuration 99
6.4 Install the WebSphere Application Server 100
6.4.1 Install WAS V3.5 101
6.4.2 Install WAS V3.5 Fixpack 2 103
6.4.3 Install WAS V3.5.2 E-fixes for WCS V5.1 104
6.4.4 Configure the WebSphere Application Server 106
6.4.5 Verify the WebSphere Application Server installation 108
6.5 Install the WebSphere Commerce Suite 108

6.5.1 Prerequisites for WCS V5.1 108
6.5.2 WCS V5.1 install 109
6.6 Create a WCS instance 111
6.6.1 Creating a WCS instance using the Configuration Manager. 111
6.6.2 Verify the WCS instance 114
6.7 Deploy the sample store. 115
6.7.1 Publish store from Store Services. 115
6.7.2 Creating an alias for the store 118
6.8 Installing a 2-tier runtime environment 120
6.8.1 Step 1 (2-tier): Install Windows NT/2000 and service packs. 120
6.8.2 Step 2 (2-tier): Install the HTTP Server. 120
6.8.3 Step 3 (2-tier): Install the DB2 Server 120
6.8.4 Step 4 (2-tier): Install the DB2 Client. 121
6.8.5 Step 5 (2-tier): Install the WebSphere Application Server. 123
6.8.6 Step 6 (2-tier): Install the WebSphere Commerce Suite 123
6.8.7 Step 7 (2-tier): Create a WCS instance. 124
6.8.8 Step 8 (2-tier): Deploy the sample store 126
6.9 Installing a 3-tier runtime environment 126
6.9.1 Step 1 (3-tier): 2-tier runtime configured 126
6.9.2 Step 2 (3-tier): Install the HTTP Server. 126
6.9.3 Step 3 (3-tier): Install the WebSphere plug-in 126
6.9.4 Step 4 (3-tier): Configure OSE on the WAS 128
6.9.5 Step 5 (3-tier) Configure OSE on the HTTP Server 129
6.9.6 Step 6 (3-tier): Verify the default host 131
6.9.7 Step 7 (3-tier): Add the WCS entries to the httpd.conf 133
6.9.8 Step 8 (3-tier) Serve static content from Web server 135
6.9.9 Step 9 (3-tier): Verify the sample store and WCS tools 135
6.10 Where to find more information 136

Chapter 7. WCS runtime for AIX: DB2 and IHS. 137
7.1 WCS runtime environment for AIX 138
7.1.1 Hardware and software prerequisites 138
7.1.2 Hardware used in our test environment 138
7.1.3 Software used in our test environment 138
7.1.4 Install AIX 140
7.2 Install the IBM HTTP Server. 144
7.2.1 Pre-installation tasks. 145
7.2.2 Install the HTTP Server. 145
7.2.3 Configure the HTTP Server. 147
7.2.4 Verify the HTTP Server. 151
7.3 Install the DB2 Server 151
7.3.1 Pre-installation requirements 152
7.3.2 Install the DB2 V7.1 Server. 155

7.3.3 Install the DB2 V7.1 Fixpack 2a 157
7.3.4 Create the WAS database on the DB2 Server 157
7.4 Install the WebSphere Application Server. 157
7.4.1 Install WAS V3.5 158
7.4.2 Install WAS V3.5 Fixpack 2. 160
7.4.3 Install WAS V3.5.2 E-fixes 161
7.4.4 Configure the WebSphere Application Server 162
7.4.5 Verify the WebSphere Application Server. 164
7.5 Install the WebSphere Commerce Suite 164
7.6 Create a WCS instance 166
7.6.1 Creating a WCS instance using the Configuration Manager. 166
7.6.2 Creating a WCS instance via the command line. 169
7.6.3 Verifying a WCS instance 172
7.7 Deploy the sample store. 173
7.7.1 Publish store from Store Services. 173
7.7.2 Deploy the sample store via the command line 175
7.7.3 Creating an alias for the store. 177
7.8 Installing a 3-tier runtime environment 179
7.8.1 Step 1: Install AIX and prerequisites. 180
7.8.2 Step 2: Install the DB2 server 180
7.8.3 Step 3: Install the DB2 client. 180
7.8.4 Step 4: Install the WebSphere Application Server 183
7.8.5 Step 5: Install the HTTP Server 183
7.8.6 Step 6: Install the WebSphere plug-in. 183
7.8.7 Step 7: Configure OSE on the WebSphere Application Server . . . 185
7.8.8 Step 8: Configure OSE on the HTTP Server. 186
7.8.9 Step 9: Install the WebSphere Commerce Suite. 188
7.8.10 Step 10: Copy the loader package 189
7.8.11 Step 11: Create the WCS database manually 190
7.8.12 Step 12: Create a WCS instance 191
7.8.13 Step 13: Add the WCS entries to the httpd.conf 198
7.8.14 Step 14: Deploy the sample store. 199
7.8.15 Step 15: Serve static content from the Web server. 202
7.8.16 Step 16: Remote administration considerations 203
7.8.17 Step 17: Configure the IPSec tunnel. 206
7.9 Where to find more information 209

Chapter 8. WCS runtime for Solaris: Oracle8i and iPlanet 211
8.1 WCS runtime environment for Solaris. 212
8.1.1 Hardware and software prerequisites 212
8.1.2 Hardware and software used for 2-tier configuration 212
8.1.3 Install Solaris 7 and patches required for WCS V5.1 213
8.2 Install the iPlanet Web Server 213

8.2.1 Install Netscape Communicator 214
8.2.2 Pre-installation steps for the iPlanet Web Server 215
8.2.3 Install the iPlanet Web Server 215
8.2.4 Enable SSL for iPlanet Web Server 218
8.2.5 Verify iPlanet Web Server 221
8.3 Install the Oracle8i Server 222
8.3.1 Oracle8i install prerequisites 222
8.3.2 Install Oracle8i Enterprise Edition Database Server 227
8.3.3 Oracle8i post install configuration 230
8.3.4 Prepare Oracle8i databases for WAS and WCS 233
8.4 Install the Oracle8i Client 241
8.5 Install the WebSphere Application Server 245
8.5.1 WAS V3.5 install prerequisites 246
8.5.2 WAS V3.5 install 246
8.5.3 Configure port 80 for the iPlanet Web Server 249
8.5.4 WAS configuration for Oracle8i 250
8.5.5 Configuring the WebSphere Application Server 251
8.5.6 WAS V3.5 install verification 252
8.5.7 WAS V3.5 Fixpack 2 installation 254
8.5.8 WAS V3.5 E-fixes installation required by WCS V5.1 254
8.6 Install the WebSphere Commerce Suite 256
8.6.1 WCS V5.1 install prerequisites 256
8.6.2 WCS V5.1 installation 256
8.7 Create a WCS instance 258
8.7.1 WCS instance creation prerequisites 258
8.7.2 Creating a WCS instance using the Configuration Manager 259
8.7.3 Post WCS instance creation configuration 262
8.8 Deploy the sample store 265
8.9 Where to find more information 267

Part 3. Developing a store 269

Chapter 9. Store archive overview 271
9.1 Store archive overview 272
9.2 Contents of a store archive 272
9.2.1 File assets 272
9.2.2 Store database assets 273
9.2.3 Descriptor 273
9.3 Physical structure of a SAR file 275
9.3.1 Standard non-multicultural store 276
9.3.2 Multicultural store 276
9.4 Creating a SAR file 277
9.4.1 Using Store Services 277

9.4.2 Building your own SAR 278
9.4.3 Publishing a SAR file 278
9.5 Understanding a Commerce Studio project 279
9.5.1 Commerce Studio file structure 279
9.5.2 Data flow between SAR and WebSphere Commerce Studio 280
9.5.3 Data flow between WC Studio and SAR 281

Chapter 10. Development environment 283
10.1 Development environment and tools overview 284
10.1.1 Development tools for customization needs 285
10.1.2 WebSphere Commerce Studio 286
10.1.3 Other development tools 288
10.1.4 Methods of installing the development environment 290
10.2 Install the development environment 291
10.2.1 Development environment hardware and software 291
10.2.2 Development environment install prerequisites 293
10.2.3 Install WebSphere Studio 293
10.2.4 Install VisualAge for Java 295
10.2.5 Install WebSphere Commerce Studio 298
10.2.6 Install XML editor 299
10.2.7 Installing other development tools 300
10.3 Configure the development environment 300
10.3.1 Configure WebSphere Studio 300
10.3.2 Configure VisualAge for Java 302
10.4 Tips for development 315
10.4.1 Stop unneeded Windows services 315
10.4.2 Disable Payment Manager 316
10.4.3 Starting Payment Manager as a Windows service 318
10.4.4 Cache tips 318
10.5 Test environments and tools 318
10.6 Where to find more information 319

Chapter 11. Create and customize a store using Commerce Studio . . . 321
11.1 Overview 322
11.1.1 Types of stores 322
11.1.2 Approaches to creating a store 323
11.2 Create a store 323
11.2.1 Create a store template 323
11.2.2 Generate a store from the template 326
11.2.3 Publish the new store 327
11.2.4 Updating VAJ WTE to run new store 327
11.3 Load store assets into Commerce Studio 328
11.3.1 Unzip SAR to PackageSAR directory 329

11.3.2 Create a Studio project 330
11.3.3 Import store assets into Studio 331
11.3.4 Studio configuration after import of assets 336
11.4 Configuring Studio for publishing 338
11.4.1 Create publishing stage 339
11.4.2 Create a server to publish to 340
11.4.3 SAR: define assets to publish and target publishing path 340
11.4.4 WCS: define assets to publish and target publishing path 342
11.4.5 WTE: define assets to publish and target publishing path 344
11.5 Publishing from Studio 346
11.5.1 Publish all assets defined for a stage and server 346
11.5.2 Publish selected files. 346
11.5.3 Creating a SAR 347
11.6 Basic customization of Web assets 347
11.6.1 Example 1: changing the store logo 348
11.6.2 Example 2: changing a label 349
11.6.3 Example 3: changing a properties files 350
11.6.4 Example 4: changing the descriptor file 351
11.7 Where to find more information 352

Chapter 12. Creating a store archive (SAR) for deployment 355
12.1 Overview 356
12.2 Publish Studio project. 356
12.2.1 Create PackageSAR directory 357
12.2.2 Importing or creating store assets in Studio 357
12.2.3 Create SAR packaging stage 358
12.2.4 Create server for SAR packaging stage 358
12.2.5 Defining assets to publish and publishing targets 359
12.2.6 Publish Studio project to PackageSAR directory 359
12.3 Packaging a SAR 359
12.3.1 Packaging a SAR from the command line using PKZIP 360
12.3.2 Packaging a SAR from a GUI zip utility. 364
12.4 Deploying a SAR 367
12.4.1 Deploy a SAR from Store Services 367
12.4.2 Deploy a SAR from command line 368
12.5 Verify store 370
12.5.1 Clear cache 370
12.5.2 Testing your store 372
12.6 Where to find more information 373

Chapter 13. Multicultural enablement 375
13.1 Overview 376
13.2 Multiculture-sensitive parts of a store 376

13.2.1 Language 376
13.2.2 Currency 377
13.2.3 Data format and representation 377
13.2.4 Address format 378
13.2.5 Catalog content 378
13.2.6 Page design 378
13.2.7 Pricing 378
13.2.8 Taxation 379
13.2.9 Shipping 379
13.2.10 Payment 379
13.3 WCS V5.1 multicultural enablement 380
13.3.1 Multicultural support for the Commerce Suite tools 380
13.3.2 Overall multicultural approach 381
13.3.3 JavaServer Pages (JSPs) 382
13.3.4 Properties files 382
13.3.5 Resource bundles 384
13.3.6 Language ID and locale 384
13.3.7 WCS V5.1 uses Unicode UTF-8 encoding 385
13.3.8 The sarinfo.xml in a multicultural-enabled store 386
13.4 Template management 386
13.4.1 One template for all stores and all languages 387
13.4.2 One template per language 387
13.4.3 One template per store 388
13.5 Multicultural programming model 388
13.5.1 Multicultural page framework 389
13.5.2 Data storage 390
13.5.3 Data input 390
13.5.4 Data output 391
13.6 Sample store multicultural features overview 391
13.6.1 Programming model 391
13.6.2 JSP include of getResource.jsp 392
13.6.3 Catalog content 393
13.7 A working example 393
13.7.1 Creating a multicultural-enabled store 393
13.7.2 Add/change language support on instance level 394
13.7.3 Translation of assets 396
13.7.4 Translation of properties files 396
13.7.5 Update the language description table 396
13.7.6 Enable counter currency/dual currency display 397
13.8 Tips for multicultural enablement 398
13.9 Where to find more information 399

Chapter 14. Implementing auctions 401

14.1 Auctions overview . . . 402
14.1.1 When to use auctions . . . 402
14.1.2 Types of auctions . . . 402
14.1.3 Auction rules . . . 403
14.1.4 Auction style . . . 404
14.1.5 AutoBids . . . 404
14.1.6 Scheduled jobs . . . 404
14.2 Create a sample auction store . . . 406
14.2.1 Enabling the sample auction store . . . 406
14.2.2 Add an auction to an existing store . . . 409
14.2.3 Enable an existing store for auctions . . . 415
14.3 Create an auction . . . 415
14.3.1 Auction wizard . . . 416
14.3.2 CreateAuction command . . . 419
14.4 Auction database tables . . . 419
14.4.1 Auction tables . . . 420
14.4.2 CleanJob . . . 420
14.4.3 Auctions and bidding data model . . . 421

Part 4. Managing a store . . . 423

Chapter 15. Deployment and catalog data management . . . 425
15.1 Deployment . . . 426
15.1.1 Deployment runtime environments . . . 426
15.1.2 Methods of store deployment . . . 427
15.2 Loader package . . . 428
15.2.1 Loader package data load process . . . 428
15.2.2 Sample data load using the loader package . . . 430
15.2.3 Loader package usage considerations . . . 434
15.3 Catalog assets in a store archive . . . 434
15.4 Where to find more information . . . 436

Chapter 16. Administrative tools, tasks, logs, and troubleshooting . . . 437
16.1 Administration tools overview . . . 438
16.1.1 WCS documentation . . . 438
16.1.2 WCS Configuration Manager . . . 439
16.1.3 WCS Store Services . . . 439
16.1.4 WCS Administration Console . . . 440
16.1.5 WCS Accelerator . . . 440
16.1.6 WebSphere Commerce Analyzer . . . 443
16.1.7 HTTP Server Administration Console . . . 444
16.1.8 WAS Administration Console . . . 445
16.1.9 DB2 Control Center . . . 445
16.2 Caching . . . 445

16.2.1 Types of caching 446
16.2.2 Configure caching 446
16.2.3 Manage caching 450
16.3 The dbclean utility 452
16.3.1 Database tables and record types 452
16.3.2 The dbclean command syntax 453
16.3.3 Using the dbclean command. 455
16.4 Job Scheduler utility 456
16.4.1 Job Scheduler commands. 456
16.4.2 Job Scheduler tables. 460
16.4.3 Using the Job Scheduler 461
16.5 Logging and troubleshooting 462
16.5.1 Enable logging and tracing 463
16.5.2 Troubleshooting using log files 466

Part 5. Back-end integration 469

Chapter 17. SecureWay Directory (LDAP). 471
17.1 Overview 472
17.1.1 What is a directory? 472
17.1.2 What is LDAP? 472
17.1.3 Why should I use LDAP? 473
17.1.4 What is IBM SecureWay Directory V3.2? 473
17.1.5 Where to get SecureWay Directory V3.2 474
17.2 WCS V5.1 LDAP support 475
17.2.1 Overview of WCS V5.1 LDAP support 475
17.2.2 Directories supported by WCS 475
17.2.3 What's new in WCS V5.1 LDAP support. 476
17.2.4 LDAP authentication scenarios 478
17.3 SecureWay Directory V3.2 installation 480
17.3.1 SecureWay Directory V3.2 runtime environment 480
17.3.2 Hardware and software prerequisites 481
17.3.3 Hardware used in our test environment 481
17.3.4 Software used in our test environment 481
17.3.5 AIX 4.3.3 and prerequisite file sets installation 483
17.3.6 IBM JDK 1.1.8 PTF 10 installation 485
17.3.7 Netscape Navigator installation. 488
17.3.8 IBM HTTP Server V1.3.12 installation 488
17.3.9 IBM DB2 V7.1 Server installation 488
17.3.10 IBM SecureWay Directory V3.2 installation 489
17.3.11 Install SecureWay Directory V3.2 489
17.4 SecureWay Directory V3.2 configuration 491
17.4.1 Configure SecureWay Directory using a GUI 491

17.5 IBM SecureWay Directory V3.21 administration 496
17.5.1 Start and stop the server 496
17.5.2 Add and delete suffixes 497
17.5.3 Add entries to the directory database 499
17.6 SecureWay Directory V3.2 SSL implementation 502
17.6.1 Prerequisites 504
17.6.2 GSKit installation 504
17.6.3 SecureWay Directory SSL configuration 505
17.7 Configuring WCS V5.1 for LDAP 509
17.7.1 WCS SSL support 513
17.7.2 Configuration verification 513
17.7.3 IBM SecureWay Directory V3.21 SSL verification 513
17.7.4 128-bit encryption security verification 513
17.8 SecureWay Directory Management Tool (DMT) 516
17.8.1 Connecting to directory servers 517
17.8.2 Administering schema object classes and attributes 518
17.8.3 Administering a directory tree 521
17.8.4 Administering directory entry ACLs 524
17.8.5 Troubleshooting 524
17.9 Working example 1 525
17.9.1 Three different scenarios 525
17.9.2 Working example prerequisites 525
17.9.3 User entry is replicated from WCS to LDAP 526
17.10 Working example 2 528
17.10.1 Three different scenarios 528
17.10.2 Working example prerequisites 528
17.10.3 User entry is replicated from LDAP to WCS 529
17.11 Where to find more information 534

Chapter 18. WebSphere Payment Manager 535
18.1 WebSphere Payment Manager overview 536
18.1.1 WebSphere Payment Manager hardware 536
18.1.2 Payment Manager installation scenarios 536
18.1.3 ITSO Payment Manager runtime scenario 537
18.2 Install Payment Manager 540
18.2.1 Install DB2 server 540
18.2.2 Payment Manager installation prerequisites 540
18.2.3 Install Payment Manager 543
18.3 Configure Payment Manager 544
18.3.1 Post installation configuration 544
18.3.2 Configure Payment Manager for a WCS Store 548
18.4 Configure WCS for Payment Manager 549
18.5 Where to find more information 551

Chapter 19. WCS messaging using MQSeries and e-mail 553
19.1 WebSphere Commerce Suite messaging 554
19.1.1 Message transports. 554
19.1.2 Messaging and queueing with IBM MQSeries 554
19.1.3 Inbound messages 555
19.1.4 Outbound messages. 556
19.1.5 High level steps to enable MQSeries messaging 556
19.2 MQSeries installation 559
19.3 Create the WCS MQSeries queues 560
19.4 JMS installation and configuration 561
19.5 WCS MQ adapter configuration 565
19.5.1 Enable the messaging system transport adapter 566
19.5.2 Update the WAS classpath variable 566
19.5.3 Configure WCS transports and messaging. 567
19.5.4 Additional settings in WCS 571
19.6 WCS MQ adapter verification 572
19.7 e-mail messaging enablement 576
19.7.1 Prerequisites for e-mail notification. 576
19.7.2 WCS e-mail transport configuration 576
19.7.3 Configuring messages for e-mail notification 577
19.7.4 Verify e-mail messaging configuration 579

Part 6. Appendixes 581

Appendix A. WebSphere Application Server tips. 583
Creating a new WAS database 584
WAS log files 584

Appendix B. Solaris tips 585
Solaris 7 installation 586
Planning for Solaris 7 installation 586
Solaris 7 base installation 587
Solaris 7 interactive installation. 589
Solaris 7 post install configuration. 591
Solaris 7 patches required for WCS V5.1 591
Common Solaris tasks and commands 593
Where to find information about Solaris 594

Appendix C. Oracle tips 595
Oracle8i commands and tasks 596
Oracle8i server 596
Oracle8i listener 596
Where to find more information 598

Appendix D. Network Dispatcher 599
WebSphere Edge Server overview 600
Network Dispatcher 600
Web Traffic Express 600
Network Dispatcher overview 600
Web server alias definition 601
Network Dispatcher alias definition 602
Network Dispatcher installation (AIX) 602
Network Dispatcher administration 604
Network Dispatcher administration using the GUI 604
Network Dispatcher graphical monitor 606
Network Dispatcher remote administration 607
Session affinity in Network Dispatcher 607
Where to find more information 608

Related publications 609
IBM Redbooks 609
Other resources 610
Referenced Web sites 611
How to get IBM Redbooks 613
IBM Redbooks collections 614

Special notices 615

Index 617

Preface

This redbook provides I/T architects, I/T specialists, and developers with the knowledge to develop, implement, and manage stores using WebSphere Commerce Suite V5.1 and WebSphere Commerce Studio V5.1.

Part 1, “Introduction to WebSphere Commerce Suite V5.1”, describes the key concepts, features, and skills required for using WCS V5.1. Next, we explore the systems architecture and programming model.

Part 2, “Implementing the runtime environment”, describes the runtime environment and provides in-depth examples on implementing 2-tier and 3-tier WCS runtimes for Windows NT/2000, Solaris, and AIX.

Part 3, “Developing a store”, describes the WCS V5.1 store development process and environment, store creation and customization using Commerce Studio, creating a SAR, multicultural enablement, and auctions.

Part 4, “Managing a store”, details the tools and tasks for store deployment, catalog and systems management.

Part 5, “Back-end integration”, provides detailed instructions for integrating SecureWay Directory (LDAP), Payment Manager, and MQSeries.

Part 6, “Appendixes”, includes tips on WebSphere, AIX, Solaris, DB2, Oracle, and Network Dispatcher.

© Copyright IBM Corp. 2001

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 IBM Redbook team (1st row: Hari Tejsingh, John Ganci, 2nd row: Michael Fritsch, Nick O'Keeffe, Paul Tanner, Michael Ambjorn, Michal Jordan-Rozwadowski was not available for the photo)

John Ganci is a Senior Software Engineer, WebSphere Specialist at the IBM ITSO, Raleigh Center. John has 14 years of experience in product and application design, development, system testing, and consulting. His areas of expertise include e-commerce, personalization, pervasive computing, and Java programming. Before joining the ITSO, he developed e-commerce sites for IBM.

Michael Ambjorn is an I/T Specialist with IBM Global Services in the United Kingdom. He has several years of experience in the e-commerce and e-business sectors as a pre-sales specialist.

Michael Fritsch is an I/T Specialist with IBM Global Services, Business Innovation Services, in Munich, Germany. He holds a degree in Business Administration from the Munich School of Management, Germany. He has several years of experience in the e-commerce and e-business sectors. His areas of expertise include e-business solutions, WebSphere Commerce, and pervasive computing. He is currently working on a project implementing WebSphere Commerce Suite V5.1.

Michal Jordan-Rozwadowski is an I/T Specialist at the Centre for IBM e-business Innovation in Toronto, Canada. His areas of expertise include e-commerce, Java programming, and professional writing.

Nick O'Keeffe is an I/T Architect at the Enterprise Application Integration Centre at IBM Hursley, in the United Kingdom. He has 25 years of I/T experience at IBM including manufacturing systems, network services and Web hosting. His areas of expertise include e-commerce and Web server scalability.

Paul Tanner is a Senior I/T Architect with IBM Global Services in Melbourne, Australia. He has nearly 20 years of experience in the I/T field and has specialized in the Internet industry over the past seven years. He holds a degree in Computer Science from Swinburne University of Technology. His areas of expertise include systems and application architecture, networking, and security.

Hari Tejsingh is a Consulting I/T Architect with the IBM Architecture Center of Excellence, East Region, US. He has seven years of experience in database and application software: architecture, design and development. He holds a Bachelor of Engineering Degree in Electrical & Electronics, M.B.A in Marketing, and a Master of Science Degree in management information systems. His current area of expertise is e-commerce solutions.

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

Peter Kovari, IBM ITSO Raleigh Center

Bill J Moore, IBM ITSO Raleigh Center

Mark Ho, IBM Toronto

Anne Marie Lafond, IBM Toronto

Siva Kumar, IBM India

Ashish Cowlagi, IBM Waltham

Bob Fraser, IBM Toronto

Asha Mishra, IBM Toronto

Jigar Desai, IBM Atlanta

Special notice

This publication is intended to help I/T architects, I/T specialists, developers and consultants who design, develop, test, deploy, and manage e-commerce Web sites using WebSphere Commerce Suite V5.1. The information in this publication is not intended as the specification of any programming interfaces that are provided by WebSphere Commerce Suite V5.1, Pro Edition, or WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

See the PUBLICATIONS section of the IBM Programming Announcement for WebSphere Commerce Suite V5.1, Pro Edition (specific platform), and WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 for more information about what publications are considered to be product documentation.

IBM Trademarks

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

e (logo)® 
IBM ®
AIX
AS/400
DB2
DB2 Universal Database
Home Director
HotMedia
MQSeries
Net.Data
OS/2
OS/390
PC 300
RS/6000
Redbooks
Redbooks Logo 
S/390
SecureWay
SP
SP1
SupportPac
VisualAge
WebSphere
Lotus
Approach
Domino
eSuite
Notes

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please 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 the address on page ii.

Part 1

Introduction to WebSphere Commerce Suite V5.1

© Copyright IBM Corp. 2001

1

Introduction

WebSphere Commerce Suite V5.1 is IBM's open architecture, Java-based server e-commerce solution.

This redbook is intended as a central reference for I/T architects, I/T specialists and developers working on e-commerce projects with WebSphere Commerce Suite V5.1. We have included detailed information and working examples for implementing multi-tiered runtime environments, store development, commerce Web site management, and back-end integration.

This chapter is organized into the following sections:

- Overview
- Functionality
- Product packaging

© Copyright IBM Corp. 2001

1.1 Overview

In order to be competitive in the global marketplace, businesses need to offer greater levels of customer service and support than ever before. When customers access a Web site today, they expect to be able to browse a product catalog, buy the products online in a secure environment, and have the product delivered to their doorstep.

WebSphere Commerce Suite V5.1 is designed to provide a complete solution for a company's electronic commerce needs. The product as a whole has evolved to meet the needs of a rapidly changing market; it is constructed to incorporate many of the concepts that are now integral to business over the Internet.

Note: Throughout the redbook, we refer to WebSphere Commerce Suite as WCS.

e-commerce

Electronic commerce or e-commerce involves doing business online, typically via the Web. The terms e-business, e-tailing, and i-commerce are often used synonymously with e-commerce. e-commerce implies that goods and services can be purchased online, whereas e-business might be used as more of an umbrella term for a total presence on the Web, which includes the e-commerce component on a Web site.

m-commerce

Simply put, m-commerce is e-commerce using a mobile device such as a mobile phone, wireless PDA, or wireless laptop. Mobile commerce or m-commerce refers to the use of mobile devices to partially or completely perform a transaction electronically from a commerce Web site for the exchange of goods and services for monetary consideration.

The key distinction between m-commerce and e-commerce at present is the use of a mobile device with a Web browser to access the commerce Web site, instead of a more traditional PC Web browser client. In the future, mobile devices will become so common for electronic commerce that there will probably not be a distinction between e- and m-commerce.

For more detailed information about m-commerce refer to *Mobile Commerce Solutions Guide using WebSphere Commerce Suite V5.1*, SG24-6171.

1.1.1 e-commerce models

As e-commerce has evolved, two major types of store models have emerged: Business-to-Consumer (B2C), and Business-to-Business (B2B). WebSphere Commerce Suite supports both of these models. Another popular e-commerce model is the auction model.

Business-to-Consumer (B2C)

The Business-to-Consumer (B2C) e-commerce store model is a publicly accessible Web site offering products for sale. It is analogous to a store on the street, where any member of the public can walk in and make a purchase. A new, unknown customer is known as a guest shopper. The guest shopper has the option of making purchases, after providing some general information about themselves to fulfill the transaction (name, address, credit card, etc.). Most B2C sites encourage users to register and become members. In doing so, the business can establish a relationship with the customer, provide better service, and build customer loyalty.

Business-to-Business (B2B)

The Business-to-Business (B2B) e-commerce store model refers to an e-commerce store specifically designed for organizations to conduct business over the Internet. The two entities are known to each other and all users are registered. B2B applications can streamline operations between businesses. For example, a retailer can place orders from a supplier B2B Web site. This type of e-commerce model greatly increases the speed and efficiency of the buying process between businesses.

Auctions

Auctions can be incorporated into B2C or B2B models. Alternatively, auction e-commerce stores can stand alone. Sites dedicated to auctions act as brokers, facilitating relationships between buyers and sellers. Auctions are a well-known way of moving surplus merchandise.

1.1.2 Concepts of e-commerce

Some of the key concepts of e-commerce include:

- Product catalog

  A product catalog on the Web is analogous to a printed catalog. Products are organized into logical groups, and the display of the products are tailored to maximize the sales of the products. Customers can browse the catalog to search for products and then place orders.

- Shopping cart

  The metaphor of a shopping cart has become widely used on the Web to represent an online order basket. Customers browse an e-commerce site and add products to their shopping cart. Shoppers proceed to the check-out, to purchase the products in their shopping cart.

- Shopping flow

  A shopping flow in the e-commerce environment is the process whereby customers browse the catalog, selects products, and purchase the products.

- User profile

  Most B2C sites try to maintain information about users. They encourage users to register. Information entered, as well as information gathered during the users' visits, form the user profile. The user profile information can be used as a powerful marketing tool, to personalize the Web site content for the the user. The personalized content can be used to filter the product catalog for only products that the customer is interested in or to implement such selling techniques as cross-selling and up-selling.

1.2 Functionality

In this section, we provide an overview of the features and functionality provided by WebSphere Commerce Suite V5.1. Some of the features are new, and some are an evolution of components that existed in previous versions. For each element of functionality, we supply a brief description and, where applicable, identify the component of the product that implements the feature.

We have grouped the functionality enhancements into the following categories:

- Business enhancements
- Technical enhancements
- Management and administration tools

1.2.1 Business enhancements

Below is an overview of the key features that WebSphere Commerce Suite V5.1 includes to help maximize the profitability of your site.

Multicultural support

Support for a culturally diverse customer base is now built in to WebSphere Commerce Suite. Stores can tailor their operations to the location or cultural preference of the user, by using the same database and Web assets (JSPs) with different language files (properties files). The key multicultural features include support for multiple languages, multiple currencies, culture-specific data (formats for dates, addresses, and numbers), different taxation rules, different shipping rules, multiple payment methods, and different prices.

Multicultural support is implemented using a combination of database enhancements and property files.

Marketing campaigns

WebSphere Commerce Suite V5.1 provides an extensive set of tools for driving the marketing objectives of an e-commerce Web site. The tools included allow the creation and organization of marketing campaigns.

The Commerce Suite Accelerator is a business intelligence system that can be used to provide information on the demographics of the users of the system. This valuable information can be used as input for the creation of marketing campaigns. The implementation of marketing campaign features is available in the Commerce Suite Accelerator. It can also be used to report on the success of marketing campaigns.

Personalization

WebSphere Commerce Suite V5.1 includes several products to implement personalization for your commerce Web site:

- Blaze Advisor V3.1 (rules-based)
- Macromedia LikeMinds V5.1 (collaborative filtering)
- WebSphere Personalization V3.5 (rules-based)

These personalization products can be used to personalize content of your Web site for the user, and provide support of a marketing campaign by implementing selling techniques, such as cross-selling, up-selling, specials, etc.

Online catalog

The WebSphere Commerce Suite V5.1 catalog subsystem provides a versatile system for storing and displaying catalog data. It includes all data and logic that relates to an online catalog: categories (also known as catalog groups), products and their attributes, and items. It also encompasses any association or relationships that exist between categories, between products, and between categories and products.

Order processing

The order subsystem provided by WebSphere Commerce Suite V5.1 allows for a robust system of order processing. It provides functionality to support the concept of shopping carts and order management. Order-processing capabilities can be extended to include quick-order or buy, scheduled orders, multiple pending orders, standing orders, and interest lists.

Member and group management

The member subsystem allows for user registration and management, as well as support for user organizations and member group management. Its abilities include user authentication, access control, views and processing tailored by user group, and session and profile management services.

Advanced member and group management tools are accessed via the Commerce Suite Administration Console.

Enhanced secure payment options

WebSphere Commerce Suite V5.1 provides the capacity for secure and customizable payment processing. Payment can be either online or offline and integrates with the merchant software systems.

IBM WebSphere Payment Manager V2.2 is the component that handles the payment processing. Once installed, it can be accessed via the Commerce Suite Accelerator or the Administration Console.

Auctions

The creation and management of auctions in WebSphere Commerce Suite V5.1 has been simplified. Commerce Suite provides tools to help create and manage auctions for your site. You can choose from one of the existing auction types (open cry, sealed bid, and Dutch auction) or create custom auction styles.

The tools for creating and managing auctions are included in the Commerce Suite Accelerator.

1.2.2 Technical enhancements

In this section we list the key technical components that WebSphere Commerce Suite V5.1 uses to maximize the technical functionality of your site.

Common server runtime

This is the framework in which commerce applications are deployed and executed. The framework uses the runtime services provided by WebSphere Application Server to maximize the efficiency of WebSphere Commerce Server applications. It is built in Java, and developers can extend the base classes to customize functionality. Highlighted below are some of the key aspects of the common server runtime:

- Data beans

  Commerce Suite provides a set of data beans to display user, catalog and order information. It also provides the capacity to extend the existing data beans for customized data, or even create new ones.

- JavaServer Pages (JSPs)

  JSPs, in conjunction with data beans, simplify the process of creating dynamic Web pages. JSPs are compiled into servlets by the WebSphere Application Server.

- Enterprise bean entity framework

  This framework allows developers to access system data without being bound by the underlying database schema. The Commerce Suite common server runtime provides entity beans to access the base schema. For customization purposes, existing entity beans can be extended, or new ones can be created.

- Systems management

  The WebSphere Commerce Server process is the focal point for your electronic commerce application. It is a multi-threaded Java process which runs the JSPs, servlets and EJBs. The WebSphere Commerce Server is integrated into the WebSphere Application Server, with each Commerce Suite instance running as a single process.

  The elements of the common server runtime can be managed using the WebSphere Application Server Administrator's Console in conjunction with the WebSphere Commerce Suite Configuration Manager.

Mobile commerce (m-commerce)

WebSphere Commerce Suite V5.1 is designed to enable you to provide access to your online stores from a number of devices, including wireless devices, such as mobile phones, wireless PDAs, and wireless laptops.

Support for mobile, or "pervasive", computing is built in the Commerce Suite database schema and access beans, allowing developers to write adapters to support specific devices.

For more information on this topic, refer to *Mobile Commerce Solution Guide using WebSphere Commerce Suite V5.1*, SG24-6171.

Support for multiple authentication techniques

Authentication in WebSphere Commerce Suite V5.1 can be done using the standard X.509 authentication, or through custom authentication.

- X.509 authentication

  Commerce Suite supports client certificate logon as a security mechanism.

- LDAP authentication

  Commerce Suite also allows the option of authenticating a customer's identity against a centralized LDAP (Light Directory Access Protocol) user registry.

The WebSphere Commerce Suite V5.1 packaging includes the SecureWay Directory Server. This is a robust LDAP directory server for security and e-business solutions.

Back-end integration messaging system

WebSphere Commerce Suite V5.1 provides support for integration with back-end systems and notification of events via messaging subsystems for inbound and outbound messaging. It allows you to set up and manage the delivery of messages generated within Commerce Suite.

A number of standard inbound messages already exist in Commerce Suite. These message allow back-end systems to create and update user registrations, update the status of orders, as well as affect the inventory or price of a product. Commerce Suite also has a standard outbound message that enables Commerce Suite to initiate order creation on a back-end system.

Developers can extend the existing messages or create new messages to increase the back-end integration capabilities of a store.

Configuration of back-end messaging is achieved by using the Configuration Manager in conjunction with the Administration Console.

Development tools

On the Windows NT and Windows 2000 platforms, the WebSphere Commerce Studio package includes a number of powerful development tools. The two key development tools are VisualAge for Java V3.5 and WebSphere Studio V3.5.

- VisualAge for Java

  This integrated development environment for Java helps reduce overall effort and development time. Two key features are the WebSphere Test Environment, which enables developers to test their Java code and JSPs locally before exporting them to the server, and the enterprise bean mapping tool, which allows for easy creation of entity beans.

- WebSphere Studio

  Using the WebSphere Studio page design tool, a Web designer can drop beans onto a JSP file to provide dynamic content with a minimum of programming effort.

Support for multiple Web servers

The WebSphere Commerce Suite V5.1 packaging includes IBM HTTP Server V1.3.12. In addition, Commerce Suite support Netscape iPlanet Web Server, Enterprise Edition V4.15 and Lotus Domino Web Server V5.0.5.

Support for multiple databases

WebSphere Commerce Suite V5.1 packaging includes IBM DB2 Universal Database V7.1, Enterprise Edition. In addition, Commerce Suite supports Oracle 8i (8.1.6).

1.2.3 Management and administration tools

WebSphere Commerce Suite V5.1 supplies a number of tools for creating, customizing, and managing the operational features of the store.

Store Services

This tool focuses on helping to create and customize stores. Users can gather all of their catalog information, Web assets (JSPs), and property files into a central location known as a store archive (SAR), and then publish the store archive to the WebSphere Commerce Server runtime environment.

Commerce Suite Accelerator

Commerce Suite Accelerator delivers facilities for managing your store and enabling your business initiatives. From within Commerce Suite Accelerator, you can perform store and product management, organize marketing campaigns, manage customer orders, and create auctions.

Administration Console

The Administration Console is a command center that offers tools for user and group management, access control, performance monitoring, and message configuration. It also furnishes an interface to the Payment Manager and the Blaze Rules Advisor.

Loader package

WebSphere Commerce Suite V5.1 supplies the loader package, which allows for loading of product and catalog information through XML files. A catalog can be created and maintained in an XML file, or exported to a file from another source and then loaded into the Commerce Suite database schema. The loader package also supports customized tables that have been added to the Commerce Suite schema.

1.3 Product packaging

Depending on your role, runtime environment hardware, and operating system, you will need different WebSphere Commerce Suite V5.1 software packaging. This section includes a listing of the WebSphere Commerce Suite V5.1 product packaging for runtime and development.

1.3.1 Runtime product packaging

The WebSphere Commerce Suite V5.1 runtime is packaged as follows by platform:

- WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000
- WebSphere Commerce Suite V5.1 Pro for AIX
- WebSphere Commerce Suite V5.1 Pro for Solaris
- WebSphere Commerce Suite V5.1 Pro for iSeries (AS/400)
- WebSphere Commerce Suite V5.1, Start for Windows NT and 2000

1.3.2 Development product packaging

- WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000

  This developer suite is used to develop stores to be deployed in the WebSphere Commerce Suite V5.1 Pro Edition for the supported runtime platforms.

- WebSphere Commerce Studio V5.1, Developer Edition for Windows NT and 2000

  This developer suite is used to develop stores to be deployed in the WebSphere Commerce Suite V5.1, Start Edition.

2

Skills planning

WebSphere Commerce Suite V5.1 provides many out-of-the-box features for developing, implementing and managing your commerce Web site. In order to fully take advantage of the features included in WebSphere Commerce Suite V5.1, we recommend that you assess the skills and roles of individuals in your organization and develop an education plan. The education can be in the form of classroom courses, online Web courses, Redbooks, product documentation, and online help.

This chapter is organized into the following sections:

- Roles and skills
- Matching skills to customization requirements
- Education
- Where to find more information

© Copyright IBM Corp. 2001

2.1 Roles and skills

When building a project team for the development of a commerce Web site it is imperative that you have team members with the appropriate skills for their given roles. In WCS V5.1 features and tools have been implemented to provide support for the separation of roles between I/T specialists and business users.

In this section we highlight the roles and skills required to develop commerce Web sites with varying levels of complexity. We have categorized the roles as follows:

- I/T specialist roles
  - Site administrator
  - Store administrator
  - Database administrator
  - Architect
  - Java programmer
  - Web designer
- Business user roles
  - Project manager
  - Merchant
  - Marketing Manager
  - Merchandising Manager
  - Customer Service Representative
  - Order Clerk

Note: Depending on the size of the Web site and organization, some of the roles may be performed by the same person. A large enterprise Web site may have several people performing the same role.

2.1.1 I/T specialist roles

Typically a team of I/T specialists is involved in the development and deployment of a store. For each of the roles we include a brief description, followed by common tasks and associated skills.

Site administrator

The site administrator installs, configures, and maintains the Commerce Suite and the associated software and hardware. The administrator responds to system warnings, alerts, and errors, and diagnoses and resolves system problems. This role typically controls access and authorization (creating and assigning members to the appropriate role), manages the Web site, monitors performance and manages load balancing tasks. The site administrator may also be responsible for establishing and maintaining several server configurations for different stages of development: test, staging, and production. This role also handles critical system backups and resolves performance problems.

Store administrator

The store administrator can create a store, manage the store assets, and update store information. The store administrator can also publish a store to the runtime environment. The person who handles store administration tasks requires familiarity with the Internet, and needs thorough knowledge of your business procedures.

Database administrator (DBA)

The database administrator (DBA) manages and maintains the Commerce Suite database. This member usually has database administrator skills for DB2 or Oracle. The DBA often is involved in advanced customization when database schema modifications are needed.

Architect

The architect is involved with the overall solution design of the commerce Web site. Once the requirements have been established, the architect decides on what customizations need to be performed and creates the application design.

Java programmer

The Java programmers are responsible for the development of WCS commands. Commands are Java classes that are used for flow control and implement business logic. They may extend existing commands or be created as new commands. In addition, the Java programmer is responsible for the customization of the store that includes data beans and EJB development.

Web designer

Web designers use Web authoring tools, such as WebSphere Studio Page Designer, to create the static HTML and JSPs for the store. Their primary role is in the development of Web assets for the presentation layer of the store. Web designers often work with the Java developers to integrate the flow control.

2.1.2 Business user roles

The business users are responsible for the overall success and profitability of the Web site. During the early stages of the project cycle, they are involved with establishing requirements and providing input for the design of the store. After the Web site is deployed, the business user roles are primarily operational. For each of the roles we provide a brief description followed by common tasks and associated skills.

Project manager

A project manager is the person responsible for managing the entire project from the inception through the deployment. Project managers have varying levels of responsibilities and authority.

Merchant

The merchant is concerned with the business side of a store's operation. This role requires detailed knowledge about the overall business. The merchant handles staffing, and works with the other roles to determine the store image, customer profile and merchandise characteristics. The merchant supervises the overall store objectives, profitability, and management, in addition to tracking the store sales.

Marketing manager

The marketing manager communicates the market strategy and brand messages to the customers. This role monitors, analyzes, and understands customer behavior. In addition, the marketing manager creates or modifies customer profiles for targeted selling, and creates and manages campaigns and promotions. Campaign event planning can be handled by a team comprising the merchant, marketing manager, and merchandising manager. Either the marketing manager or the merchandising manager projects the sales for a promotional event and analyzes its effectiveness.

Merchandising manager

The merchandising manager needs to understand the marketing strategy and the way the store's customers shop. The merchandising manager determines the best way to display, price, and sell products in the online store. In addition, the merchandising manager traces customer purchases and determines discounts, auctions, and suggestive selling techniques. This role also supervises catalog administration, which includes creating and managing online product catalogs, associated pricing schemes, product categories, and product advising. This person is typically an expert on the product domain with a thorough understanding of the product line and the relationships between various products.

Customer service representative

No matter how well your online store is designed to provide a customer with self-service features, there will be occasions when a customer requires a personal contact. The customer service representative handles customer inquiries, whether through e-mail, fax, or phone. Some common inquiries for e-businesses include resetting passwords, updating information, or checking an order's status. Customer service representatives also deal with customer registration, and withdrawing bids and managing forum discussions for auctions on behalf of the customer.

Order clerk

The order clerk manages order processing, ensuring that orders are properly filled, payment is received, and orders are shipped. The order clerk can search for customer orders, view details, and manage order information.

2.2 Matching skills to customization requirements

A commerce store can have varying levels of customization. The skills required depend on this level of customization. The level of customization affects both the development time and skills needed to develop a store. We have categorized the customization requirements into three levels: basic, intermediate, and advanced. For each level we describe the tasks involved and list the I/T specialists roles to complete the tasks.

2.2.1 Basic customization

Basic store customization normally involves the following:

- Catalog creation

  This involves creating a catalog and dividing the catalog into logical catalog groups or categories. The store's products, known in WCS V5.1 as catalog entries, are then assigned to catalog groups.

- Web page design

  Web page design involves the creation of graphics, static HTML pages, and JSPs for a customized look and feel for the store using the existing WCS commands.

The I/T specialist roles include site administrator, store administrator, and Web designer.

2.2.2 Intermediate customization

Intermediate store customization includes the elements of a basic customization plus the following tasks:

- Custom command development

  Custom commands, which are Java classes, are used for flow control and implement business logic. They may extend existing commands or be created as new commands. WCS commands are developed in VisualAge for Java, provided in the WebSphere Commerce Studio.

- Custom data bean development

  The development of data beans is required to represent dynamic content to be displayed in JSPs. The dynamic content is populated at runtime most often from the database. Data beans are developed in VisualAge for Java, provided in the WebSphere Commerce Studio.

The I/T specialist roles include site administrator, store administrator, Web designer, and Java programmer.

2.2.3 Advanced customization

Advanced store customization includes the elements of an intermediate customization plus the following tasks:

- Database customization

  The alteration of the existing database schema may be needed to meet the specialized needs of a business.

- EJB development

  The development of persistent Java objects to represent data in the database. For example, if you need to add a table to the database you will need to create an EJB to query and update that table. EJBs are developed in VisualAge for Java, provided in the WebSphere Commerce Studio.

- Data access bean development

  The data access bean is developed in Java and uses EJBs to access data from the database. Data access beans are developed in VisualAge for Java, provided in the WebSphere Commerce Studio.

The I/T specialist roles include site administrator, store administrator, Web designer, and Java programmer.

2.3 Education

In this section, we outline the main sources of education provided by IBM Learning Services that enable you to augment the skills required for WCS based on the roles identified above.

IBM Learning Services provides education for WebSphere Commerce Suite V5.1 and related topics in a traditional classroom setting and in some cases online via Web books. The courses can be taken individually or as part of a roadmap to broaden your skills for your specific role.

This section is organized as follows:

- WebSphere Commerce Suite V5.1 courses
- WebSphere and VisualAge for Java courses
- Education roadmaps

2.3.1 WebSphere Commerce Suite V5.1 courses

This section is intended to highlight the key courses provided by IBM Learning Services on WebSphere Commerce Suite V5.1, as seen in Table 2-1.

Table 2-1 WebSphere Commerce Suite V5.1 courses

Course Number	Course name	Role and skill
WA01A	WebSphere Commerce Suite V5.1 Overview (download and play)	Site administrator, project manager
WA30A	WebSphere Commerce Suite V5.1 Store Design	Web designer, Java programmer, architect
WA10A	Planning your e-commerce Site using WebSphere Commerce Suite V5.1	Project manager, architect, lead I/T specialists
WA35A	WebSphere Commerce Suite Store Programming	Java programmer, other I/T specialists
WA20A	System Administration using WebSphere Commerce Suite V5.1	Site, store, and database administrator

The most current information on education courses and roadmaps offered by IBM Learning Services for WebSphere Commerce Suite V5.1 can be found at the following URL:

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

2.3.2 WebSphere and VisualAge for Java courses

This section is intended to highlight the key courses provided by IBM Learning Services on WebSphere and VisualAge for Java. The information taught in the courses outlined in Table 2-2 provides a solid foundation for WebSphere and Java 2 Enterprise Edition (J2EE) programming required for intermediate and advanced customization of WCS stores.

Table 2-2 WebSphere and VAJ courses

Course Number	Course Name	Skill	Role
N4420	WebSphere Studio Workshop (3 day, classroom)	Presentation Web page development	Web designer
N4430	WebSphere Application Server Development and Studio (5 day, classroom)	Presentation and flow control	Java programmer
AD63A	WebSphere Application Server AE and EJB workshop (5 day, classroom)	Flow control and business logic	Java programmer

2.3.3 Education roadmaps

This section provides education roadmaps for e-business and application development. Refer to the following URL for more detailed information:

http://www.ibm.com/services/learning/roadmaps/adotidx.html

Roadmaps for WebSphere Commerce Suite

IBM Learning Services has defined education roadmaps based on project phase and role of the I/T specialists or business users for WCS V5.1 as follows:

- Implementors

  http://www.ibm.com/services/learning/roadmaps/adebus09.htm

- Customizers

  http://www.ibm.com/services/learning/roadmaps/adebus10.htm

- Planners

  http://www.ibm.com/services/learning/roadmaps/adebus11.htm

- Business users

  http://www.ibm.com/services/learning/roadmaps/adebus12.htm

We have provided a cross-reference of the roles and skills defined in 2.1, “Roles and skills” on page 16 with the IBM Learning Services education in Table 2-3.

Table 2-3 WCS V5.1 roadmaps - rolls and skills

IBM Learning Services education roadmaps	WebSphere Commerce Suite roles and skills
Implementors	Site administrators
	Store administrators
	Database administrators
	Java programmers *
	Web designers *
Customizers	Java programmers
	Web designers
Planners	Project manager
Business Users	Business users
Note: * Not a primary role, but good for background understanding of the product.	

Roadmaps for WebSphere and VisualAge for Java

- Architecting and Developing Applications with WebSphere Advanced Edition

  http://www.ibm.com/services/learning/roadmaps/ad04b.htm
- Architecting and Developing Applications with VisualAge for Java

  http://www.ibm.com/services/learning/roadmaps/ad03.htm

2.4 Where to find more information

In this section, we provide a listing of the major sources of information for learning WebSphere Commerce Suite V5.1 and related products. The section is organized into the following topics:

- Product documentation and online help
- IBM Redbooks
- IBM Learning Services
- IBM certifications

2.4.1 Product documentation and online help

This section includes information for finding online help and PDF documentation for WebSphere Commerce Suite V5.1, and WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

WebSphere Commerce Suite V5.1

- Online help

  WebSphere Commerce Suite V5.1 online help can be found by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Documentation**.

- PDF documentation

  The PDF files can be found as follows:

  - Windows: <Commerce_Suite_Install_Path>\docs\<locale>
  - AIX: /usr/lpp/CommerceSuite/docs/<locale>
  - Solaris: /opt/WebSphere/CommerceSuite/docs/<locale>

Table 2-4 provides a listing of the WCS V5.1 product documentation PDFs.

Table 2-4 WCS V5.1 product documentation PDFs

WCS PDF filename	Role and skill
CustomerServiceRepresentative.pdf	Customer service representative
FundamentalsGuide.pdf	I/T specialist users
MerchandisingAndMarketingManager.pdf	Merchandising and marketing manager
OrderClerk.pdf	Order clerk
ProgrammersGuide.pdf	Java programmer, Web designer, other I/T specialists
SiteAdministrator.pdf	Site administrator
StoreDeveloperSAR.pdf	Java programmer, Web designer
StoreDeveloperStoreServices.pdf	Java programmer, Web designer, site administrator
WCSuiteInstallGuidePro.pdf	Site administrator, database administrator, other I/T specialists
WhatsNew.pdf	All

WebSphere Commerce Studio V5.1

- Online help

WebSphere Commerce Studio V5.1 online help can be found by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Documentation**.

- WebSphere Commerce Studio V5.1, Install Guide (WCStudioInstallGuideProDev.pdf) can be found in the Commerce Studio <CD_ROM>\docs\<locale> directory.

2.4.2 IBM Redbooks

e-commerce Redbooks

- *WebSphere Commerce Suite V5.1 Handbook*, SG24-6167
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174
- *Mobile Commerce Solution Guide using WebSphere Commerce Suite V5.1*, SG24-6171
- *WCS V5.1 Performance Tuning,* SG24-6258
- *e-Commerce Patterns using WebSphere Commerce Suite, Patterns for e-business Series*, SG24-6156
- *Building e-commerce Solutions with Net.Commerce: A Project Guidebook*, SG24-5417

Related WebSphere Redbooks

- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153
- *WebSphere Edge Server: Working with Web Traffic Express and Net Dispatcher*, SG24-6172
- *Programming with VisualAge for Java V3.5*, SG24-5264
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755
- *Patterns for e-business: User-to-Business Patterns for Topology 1 and 2 using WebSphere Advanced Edition*, SG24-5864
- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *WebSphere Personalization Solutions Guide*, SG24-6214

2.4.3 IBM Learning Services

There are a number of courses and roadmaps available from IBM Learning Services on WebSphere Commerce Suite V5.1 and related topics. Refer to the following URL for more detailed information:

http://www-3.ibm.com/services/learning/roadmaps/adotidx.html

There are a number of Web book courses available from IBM Learning Services which can be accessed online using a Web browser. In most cases these courses are prerequisites to attending classroom courses. Refer to the following URL for more detailed information:

http://learn.ibm.be/webbooks

2.4.4 IBM certifications

This section provides URLs for certifications offered by IBM to help develop skills and marketability of your I/T specialists.

Benefits of getting certified

http://www.ibm.com/certify/program/benefits.shtml

Core skills

- IBM Certified for e-business - Solution Advisor

  http://www.ibm.com/education/certify/certs/ebcpsadv.phtml

- IBM Certified for e-business - Solution Designer

  http://www.ibm.com/education/certify/certs/ebcpsdes.phtml

- IBM Certified for e-business - Solution Technologist

  http://www.ibm.com/certify/certs/ebcpstec.shtml

e-commerce skills

- IBM Certified Specialist - IBM WebSphere Commerce Suite V4.1 Implementation

  http://www.ibm.com/software/ad/certify/adcswc4i.html

- IBM Certified Solutions Expert - IBM WebSphere Commerce Suite V4.1 Customization

  http://www.ibm.com/software/ad/certify/adsewc4c.html

Note: At the time of writing this redbook, the WCS V5.1 certifications were being developed.

Integration skills

- IBM Certified Specialist - MQSeries

  http://www.ibm.com/certify/certs/mqcsmqs.shtml

- IBM Certified Solutions Expert - MQSeries

  http://www.ibm.com/certify/certs/mqsemqs.shtml

- IBM Certified Developer - MQSeries

  http://www.ibm.com/certify/certs/mqcdmqs.shtml

- IBM Certified Specialist - MQSeries Integrator V2

  http://www.ibm.com/certify/certs/mqcsmiv2.shtml

- IBM Certified Solutions Expert - MQSeries Workflow

  http://www.ibm.com/certify/certs/mqsewkfl.shtml

- IBM Certified Developer - XML and Related Technologies, V1

  http://www.ibm.com/software/ad/certify/adcdxmv1.html

Development skills

- IBM Certified Specialist - IBM VisualAge for Java, Professional Edition, V3

  http://www.ibm.com/software/ad/certify/adcsjvv3.html

- IBM Certified Solution Developer - IBM VisualAge for Java, Professional Edition, V3

  http://www.ibm.com/software/ad/certify/adsdjvv3.html

- IBM Certified Solutions Expert - IBM WebSphere Studio, V3.5

  http://www.ibm.com/software/ad/certify/websphere/v35/adsewsv35.html

WebSphere skills

- IBM Certified Specialist - IBM WebSphere Application Server V3.5, Standard Edition

  http://www.ibm.com/software/ad/certify/websphere/v35/adsewsv35.html

- IBM Certified Solution Developer - IBM WebSphere Application Server, Standard Edition, V3.

  http://www.ibm.com/software/ad/certify/websphere/v35/adsdwsv35.html

- IBM Certified Enterprise Developer - IBM WebSphere Application Server, Advanced Edition, V3.5

  http://www.ibm.com/software/ad/certify/websphere/v35/adedserv35.html

- IBM Certified Systems Expert - Administration for IBM WebSphere Application Server, Advanced Edition, V3.5

  http://www.ibm.com/software/ad/certify/websphere/v35/adcseserv35.html

- IBM Certified Solutions Expert - Business Intelligence

  http://www.ibm.com/certify/certs/dbsebi.shtml

Database skills

- IBM Certified Solutions Expert - DB2 UDB V7.1 Family Application Development

  http://www.ibm.com/certify/certs/dbseudd7.shtml

- IBM Certified Solutions Expert - DB2 UDB V7.1 Database Administration for UNIX, Windows and OS/2

  http://www.ibm.com/certify/certs/dbseuda7.shtml

Security skills

- Tivoli Certified Solutions Expert - IBM SecureWay Firewall for Windows NT

  http://www.tivoli.com/services/certification/roadmap/cert_sway_firewall_winnt.html

- Tivoli Certified Solutions Expert - IBM SecureWay Firewall for AIX

  http://www.tivoli.com/services/certification/roadmap/cert_sway_firewall_aix.html

3

Runtime architecture

When planning for ane-commerce Web site, it is important to have a clear picture of the desired runtime architecture for the production environment of the store.

This chapter describes the business considerations for runtime architectures, runtime configurations, runtime architecture components, and the runtime computing flow. The terms runtime architecture and systems architecture are synonymous in the context of this redbook. The concepts in this chapter provide a foundation for the WCS V5.1 implementation chapters for the Windows NT and 2000, Solaris, and AIX platforms.

This chapter is organized into the following sections:

- Business considerations
- Runtime configurations
- Runtime architecture components
- Runtime computing flow
- Where to find more information

© Copyright IBM Corp. 2001

3.1 Business considerations

The final design of a runtime architecture should be the result of consideration of the stated business requirements and objectives. The technical solution selected needs to fulfill the stated business objectives. In addition, there are many considerations that are included in the technical solution that are common for nearly every installation. For example, performance and scalability may not be mentioned in the business objectives; however, the technical solution must consider these issues.

3.1.1 Scalability

Scalability is a key consideration for a Web site. It can be difficult to predict with any degree of accuracy: how many users will access the Web site, and what the connection rate and loads will be. This does not mean you should not try to estimate what type of load your Web site will encounter. By implementing a scalable runtime architecture, the infrastructure can scale as the demands of your Web site increase, without having to rewrite your application. Enhancing the scalability of the runtime architecture can be done rapidly by adding component resources as needed.

Vertical scaling

Vertical scaling is implemented by increasing the resources on a single machine or node. Increasing the resources of a single node include substituting a more powerful machine, installing more memory, and cloning software processes.

Horizontal scaling

Horizontal scaling is implemented by installing multiple physical servers of the same type. For example, multiple low-end Web servers may be used in a cluster as the front-end to a Commerce Server.

3.1.2 Availability

A commerce Web site will often operate at nearly 100% availability. This type of environment requires a runtime architecture design with high availability. For example, the components of the runtime architecture will need to be duplicated with the capability for hot backup to avoid a single point of failure.

3.1.3 Security

It will always be a requirement that an e-business site should be protected from intrusion and that all online assets are secure. Security has to be addressed at all levels, including physical building access control, operating system security, and network security. From the viewpoint of a runtime architecture, security considerations need to include all the distributed runtime architecture nodes such as database servers, and the site must be secured by one or more firewalls.

3.1.4 Back-end integration

A site can be implemented in a stand-alone manner, referred to as *Web-up*, with little or no automated integration with existing business applications if that is the objective. Very often, however, there is a need for the Web site to be fully integrated with the enterprise's legacy applications, referred to as *enterprise-out*.

In many cases, sites that are started as simple stand-alone operation eventually progress as requirements emerge for a greater degree of integration with other business systems.

3.1.5 Skills

A wide range of skills is required to support an e-commerce Web site (refer to Chapter 2, "Skills planning" on page 15). A sophisticated runtime architecture will require an industrial strength I/T operation to support the operation. This includes actions such as database backup and recovery, monitoring and alerting, security support, etc. Having the appropriate skills over a 24x7 operation may lead to a decision to outsource certain supporting staff positions for the production Web site, to organizations that provide 24x7 coverage.

3.1.6 Cost

Investments in I/T infrastructure to achieve high availability are subject to the law of diminishing returns. That is to say, each additional fractional percentage point of availability will cost progressively more in terms of the resources that have to be applied. Therefore, a risk assessment will need to be performed to help decide what is the appropriate level of investment. The cost to the business of any outages will need to be compared with the cost to achieve greater availability.

3.2 Runtime configurations

Depending on business requirements, the hardware and software components may be deployed in a variety of topologies and on a number of software platforms. For example, various components may be distributed across multiple physical server nodes and possibly combined with additional specialized components such as:

- Firewalls
- Load balancers
- Back-end application integration servers
- Directory servers

Individual nodes may be replicated or cloned to provide resilience and load distribution.

Despite the theoretically large number of possible configurations, business drivers tend to be common across enterprises. This leads to a recurring set of standard deployment architectures or patterns.

A basic question is whether to choose a single box (1-tier) installation or whether the commerce components should be distributed across separate physical machines (2-tier or 3-tier configurations).

3.2.1 Single-tier runtime configuration

A single-tier (1-tier) configuration consists of all the runtime components installed on a single physical machine. This is easiest to set up and would be suitable as a development or test environment. It would not, in most cases, satisfy the scalability, security and availability requirements of a production site unless you are using a high-end IBM zSeries or iSeries server.

The only way to scale with this type of configuration is to scale vertically by adding memory, disk space, processors to the server, and in the most extreme cases, upgrading the server to a more powerful system. A simple 1-tier system is shown in Figure 3-1.

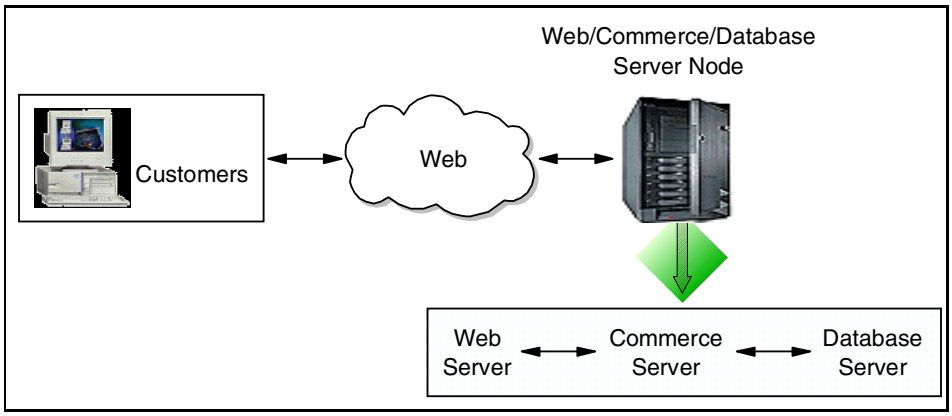


Figure 3-1 WCS 1-tier - high level view

The 1-tier configuration involves all software components residing on a single machine as seen in Figure 3-2.

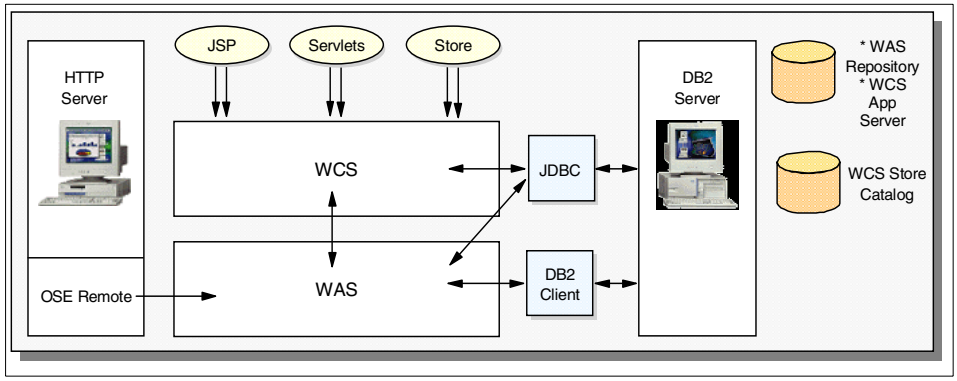


Figure 3-2 WCS 1-tier - detailed view

3.2.2 Two-tier runtime configuration

In the two-tier (2-tier) configuration, the database is installed on a separate database server node, as seen in Figure 3-3.

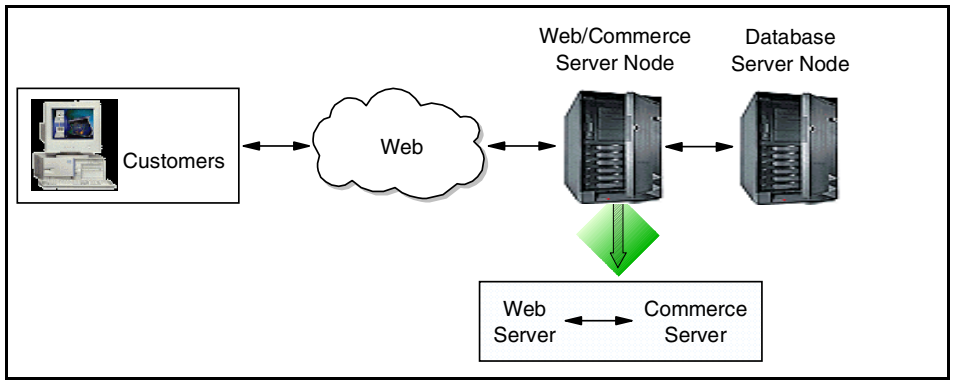


Figure 3-3 Commerce site: 2-tier configuration

This configuration will potentially achieve higher performance than the 1-tier system and also has the following advantages:

- Security

  There is an option to install a firewall between the Commerce Server and the database server. If the firewall is properly configured, this will ensure that only the Commerce Server can make connections to the database machine from the insecure side of the firewall.

- Scalability

  With the Commerce Server and database functions separated, greater scalability can be reached individually. Additional resources can be installed to address the area that may be experiencing load problems. For example, if the database machine is overloaded, it can be scaled up either vertically by installing more memory, or horizontally by implementing a high-availability database cluster. If the Web server or Commerce Server is the problem, then this can be scaled independently of the database server.

Figure 3-4 shows the 2-tier runtime environment. This environment consists of the Web Application Server (WAS) and the WebSphere Commerce Suite software running on one machine with the database residing on a separate database machine. The DB2 client component is used for administrative actions with the database server, while the JDBC driver is used for data access from the WCS/WAS applications.

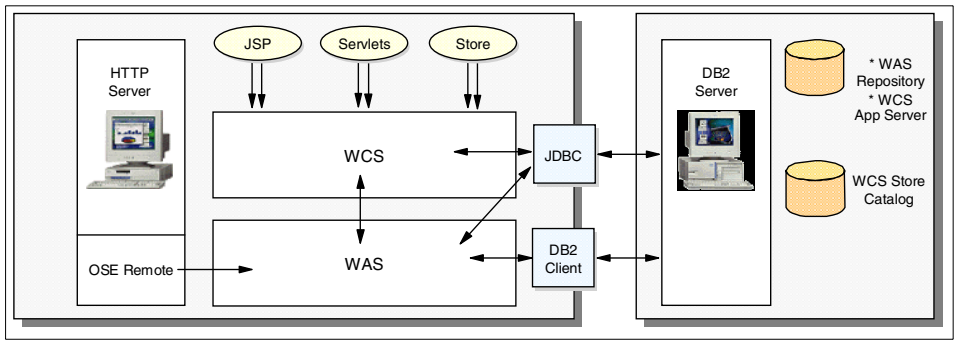


Figure 3-4 WCS 2-tier - detailed view

3.2.3 Three-tier configuration

The three-tier (3-tier) configuration separates the Web server, Commerce Server and database server nodes, as shown in Figure 3-5.

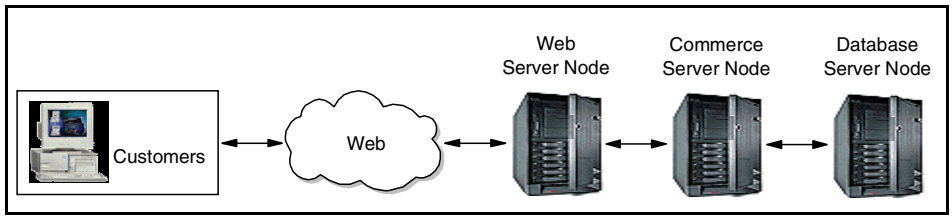


Figure 3-5 WCS 3-tier - high level view

This configuration gives a high degree of flexibility and scalability. It allows any of the nodes to be independently scaled. For example, it may be advantageous to use horizontal scaling of the Web server node to install two or more Web servers in a cluster using a load balancer to distribute requests to the servers in the cluster.

Figure 3-6 shows the 3-tier environment. This environment consists of independent machines for each of the services: HTTP, Web Application Services, and Database services. This allows for increased scalability and therefore improved performance of the deployed store, as the volume of transactions on the store increases.

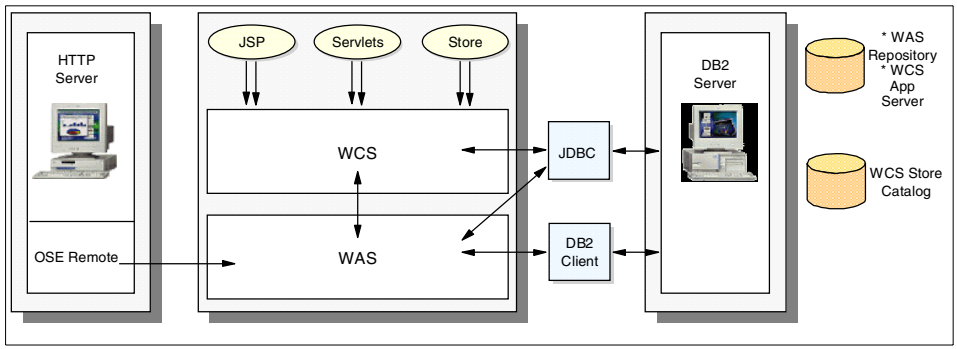


Figure 3-6 WCS 3-tier - detailed view

3.2.4 Enterprise 3-tier configuration

The 3-tier configuration allows installation of a firewall between the Web server node and the Commerce Server.

Figure 3-7 on page 37 illustrates a possible enterprise-level implementation of a 3-tier configuration. In this diagram the Web servers are shown placed in a demilitarized zone (DMZ) between two firewalls. This allows HTTP and HTTPS traffic to be routed to the Web servers. No other machines are visible (routable) from the Internet. Only the Web servers can connect through the inner firewall to the Commerce Servers.

Two Web servers are shown with load balancing of HTTP requests being implemented via the IBM Network Dispatcher, which is shown in a high-availability configuration with a standby dispatcher. The Commerce Server nodes are also horizontally scaled with Java application servers cloned across multiple machines.

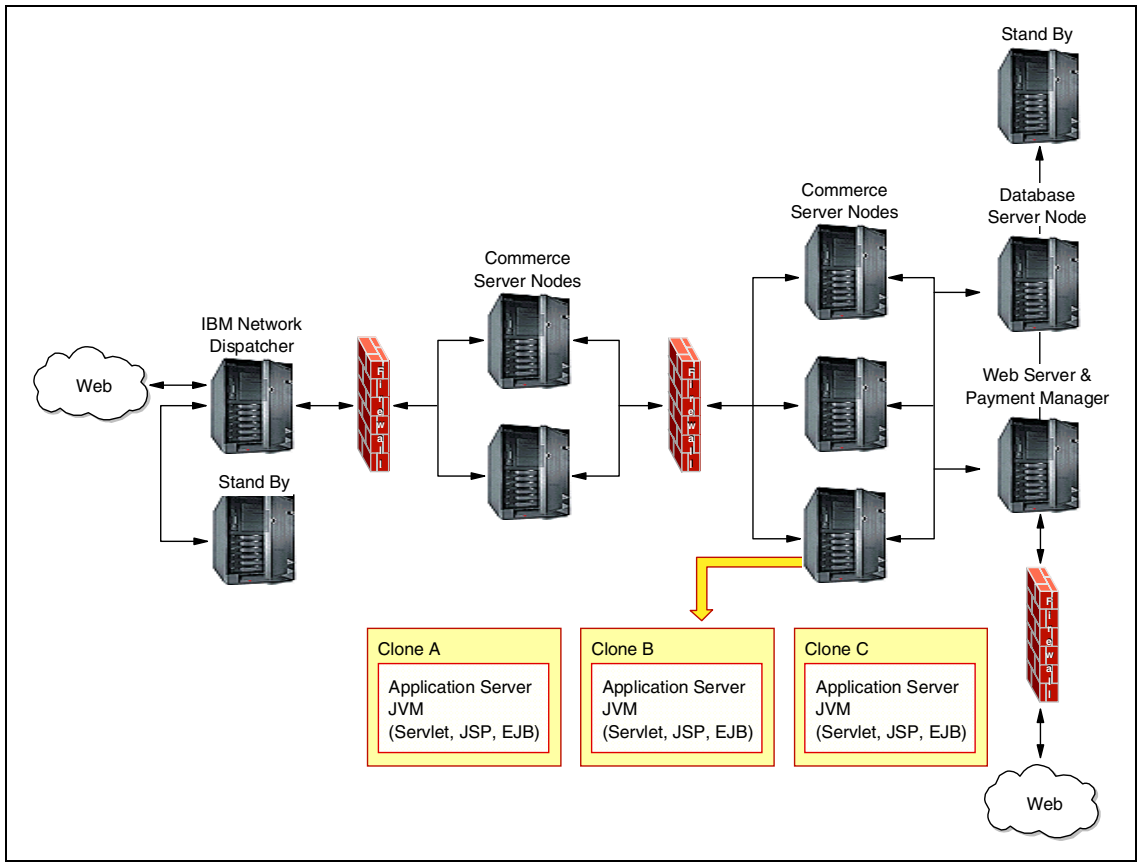


Figure 3-7 Commerce site: enterprise configuration

3.3 Runtime architecture components

An e-commerce application integrates a set of hardware and software elements to provide the function of an online store.

The typical components of a runtime architecture are as follows:

- Server hardware and operating system

  For example, IBM @server xSeries 220 with Windows NT 4.0 Server

- Web server

  For example, IBM HTTP Server V1.3.12

- Database server

  For example, IBM DB2 Universal Database V7.1, Enterprise Edition for Windows
- Application server

  For example, IBM WebSphere Application Server V3.5, Advanced Edition
- Commerce Server

  WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000
- Application code

  WCS V5.1 commerce application (store)

The configuration of these elements into an operational system is referred to here as a runtime architecture or runtime environment. A runtime environment is concerned with all aspects of the design of a system to run in a production environment.

3.3.1 Software components

To better understand the systems architecture, it is useful to examine the interaction of the main software components of the system. The main components are Web server, WebSphere Application Server, WebSphere Commerce Server, and the database server, as seen in Figure 3-8.

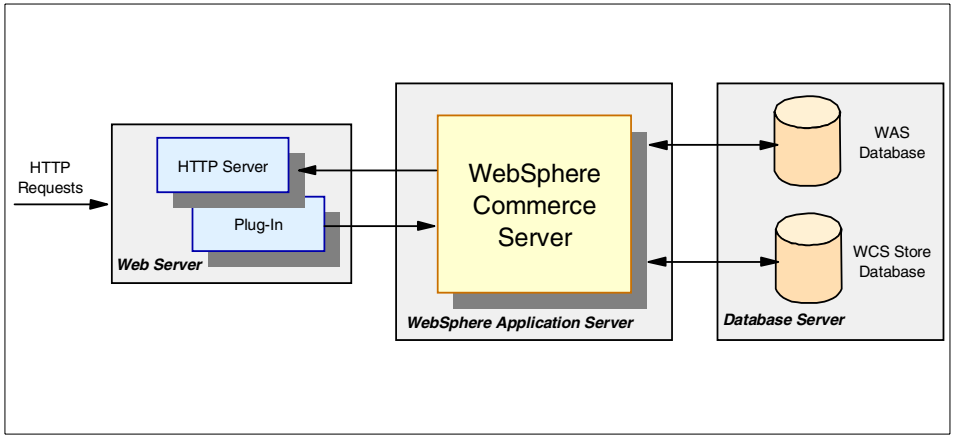


Figure 3-8 Software component interaction

Web server

WebSphere Commerce Suite V5.1 comes with IBM HTTP Server V1.3.12; however, Domino Web server and Netscape iPlanet Web server are supported. The Web server is the first point of contact for an incoming HTTP request. In order for the Web server to interface with WebSphere Application Server, the WebSphere Application Server plug-in must be installed.

WebSphere Application Server

The application server for WebSphere Commerce Suite V5.1 applications is the IBM WebSphere Application Server V3.5, Advanced Edition, known as WAS. Commerce Suite leverages the features of the WebSphere Application Server to maximize scalability, performance, and the J2EE programming support.

WebSphere Commerce Server

This is the focal point for electronic commerce applications built in WebSphere Commerce Suite V5.1. It is a multi-threaded Java application that runs as a single process within the WebSphere Application Server.

Database server

WebSphere Commerce Suite V5.1 ships with the IBM DB2 Universal Database V7.1, Enterprise Edition. It also supports the Oracle 8i (8.1.6) database server. The database server holds all of the application data, product and user information, and operational data, such as command and view registrations.

3.3.2 Commerce Suite runtime architecture components

This section provides a simplified view of the runtime architecture for a WebSphere Commerce Suite V5.1 application. The main components of the Commerce Suite runtime architecture are as follows:

- Servlet engine
- Protocol listeners
- Adapters
- Web Controller

The application is much more complex than this. Each of these components could be described in much more detail. Here we provide a simplified view to facilitate understanding of the process as it pertains to the runtime as a whole.

For a more detailed examination of this topic, please refer to the online documentation or the product guides found on the product CD:

- *Fundamentals, IBM WebSphere Commerce Suite V5.1*
- *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*

Servlet engine

The servlet engine is essentially a request dispatcher. It manages a pool of threads to handle requests and assigns each inbound request to a thread.

Protocol listeners

These are components that receive inbound requests from transports and dispatch them to the appropriate adapter. There is a listener for each type of request that a site is equipped to handle. Two common listeners are the request servlet, which handles incoming HTTP requests, and the MQSeries listener, which handles inbound XML-based MQSeries messages from remote programs (such as back-end systems).

Adapters

Adapters are a form of interpreter. They transform a request so that it can be correctly handled by the Commerce Suite commands. For example, an adapter may add special instructions to a message. These instructions are interpreted by the Web Controller, enabling it to treat the request properly. Adapters can also map data elements of an incoming message to data elements that are understood by Commerce Suite commands. Another capability of adapters is to perform device-specific session management.

Web Controller

The Web Controller is the key element of the Commerce Suite programming model. It determines which commands should be run for a given URI and, when the command is complete, which view (most often a JSP) to use to display the results.

It also simplifies the work of the Commerce Suite commands (servlets) by performing standard administrative tasks such as:

- Session management
- Transaction control
- Access control
- Authentication

It also enforces the Commerce Suite programming model.

For more information about the Web Controller and the Commerce Suite programming model, please refer to Chapter 4, “Programming model” on page 43.

3.4 Runtime computing flow

Understanding the computing flow of an HTTP request in the runtime environment can serve to better understand the interaction of the components. Figure 3-9 depicts the path taken by a typical HTTP request from a simplified view.

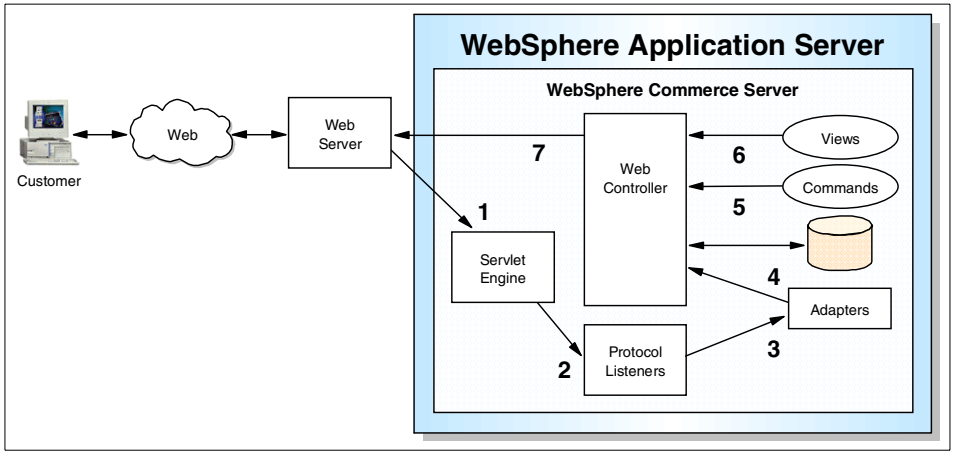


Figure 3-9 Runtime computing flow of a HTTP request

For example, a customer browsing your commerce Web site wishes to view a product category in the online catalog. The request to view the page is sent from the client Web browser to the Web server. The Web server accepts the request, and then uses the WebSphere Application Server plug-in to direct the request to the servlet engine. The remaining steps in the computing flow, as seen in Figure 3-9, are documented as follows:

1. The servlet engine is the first point of contact to the WebSphere Commerce Server. The servlet engine assigns the request to its own thread, and dispatches the request to the correct protocol listener.
2. The protocol listener in this example is the HTTP request servlet. The request servlet passes the request on to the HTTP adapter manager.
3. The adapter manager determines that the request came from a PC browser client, and invokes the appropriate adapter, in this case the HTTP browser adapter.
4. The adapter performs little function in the case of an HTTP request from a PC browser. Adapters can be written to handle the more complex cases. The HTTP browser adapter passes the request on to the Web Controller.
5. The Web Controller determines the correct command to invoke based on the uniform resource identifier (URI), by querying the WCS database. In this

example, the customer requested a category display page, so the category display command is invoked. The command performs some processing, and then returns a view name to the Web Controller.

6. The Web Controller determines the correct JSP to display for the view and the required parameters, by searching the database VIEWREG table.
7. The JSP writes a response, which is returned to the Web server. The Web server in turn relays that response to the client for the display of the category display page.

3.5 Where to find more information

For further information on the architecture of WebSphere Commerce Suite V5.1 and WebSphere scalability, refer to the following books:

- *WebSphere Commerce Suite Fundamentals Version 5.1* found on the product CD.
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition,* SG24-6153.

4

Programming model

Possibly the most significant enhancement in WebSphere Commerce Suite V5.1 is that the WebSphere Commerce Server has been written to run as a WebSphere Application Server application. The departure from the older CGI architecture to an open Java standard, provided by the WebSphere Application Server, offers tremendous advantages for the application programmer.

In this chapter, we provide an overview of the WebSphere Commerce Suite V5.1 programming model. The objective is to describe the programming model in broad strokes, to give developers a feel for what is entailed in developing a WebSphere Commerce Suite V5.1 application.

This chapter is organized into the following sections:

- Concepts
- Application development overview
- Web Controller
- Commands
- Data beans
- Views
- Error handling and messages
- Debugging methods

© Copyright IBM Corp. 2001

4.1 Concepts

There are many new concepts at the heart of the WebSphere Commerce Suite V5.1 programming model, which shape the approach of the application developer when developing commerce applications with WebSphere Commerce Suite V5.1. This section outlines the key concepts and technologies of the programming model.

4.1.1 Java 2 Platform, Enterprise Edition (J2EE)

Java 2 Platform, Enterprise Edition (J2EE) consists of a number of key components for Java server programming that now form the core of the WebSphere Commerce Suite V5.1 programming model.

For more information on Java server programming and WebSphere Commerce Suite V5.1, refer to the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1* found on the product CD.

For more information on J2EE, refer to the following URL:

`http://java.sun.com/j2ee/`

Enterprise JavaBeans (EJBs)

The Enterprise JavaBeans (EJBs) technology, referred to as beans, provides an object layer for accessing the persistent data in the WCS database.

There are two sets of EJBs in WebSphere Commerce Suite, private and public. The private beans are used by the WebSphere Commerce Suite runtime and administrative tools. The public beans can be used by the application developer as is or extended when developing business logic applications for the store.

WebSphere Commerce Suite uses both session and entity beans. The session beans perform intensive database operations, usually within the scope of a single transaction or session with the client. The entity beans represent the underlying data in the form of Java objects. They form an object layer that is persistent across multiple transactions. For example, the catalog bean provides an object model for the logical catalog entity within the store.

Once the EJBs are developed by the Java developer, they are deployed on the WebSphere Application Server in the EJB container. One of the benefits of using EJBs is that the WebSphere Application Server is capable of providing scalability. By using EJBs, each reference to the commerce application data may not necessitate a direct connection to the database, thus enhancing performance by minimizing database access. All data within the WebSphere Commerce Suite database is accessed by using the underlying Enterprise JavaBeans technology.

Java Servlets (servlets)

Java Servlets, referred to as servlets, provide the basis for server-side operations in WebSphere Commerce Suite V5.1. Servlets are organized and managed by WebSphere Application Server. This is a departure from the Net.Commerce CGI model for server-side operations.

The servlets themselves are persistent within the WebSphere Application Server, Advanced Edition environment, unlike the CGI environment where each script runs as a stand-alone entity and has a life span of only one request. Servlets also support the notion of a user session, where persistent data can be stored spanning multiple requests.

JavaServer Pages (JSPs)

JavaServer Pages, referred to as JSPs, are used by WebSphere Commerce Suite V5.1 for the presentation or view of the commerce application. Although JSPs are technically capable of performing some processing, we recommend they be used only for display (view) purposes in conjunction with using data beans to retrieve data from the WCS database. This approach is consistent with the WebSphere Commerce Suite model-view-controller model.

XML

Extensible Markup Language (XML) forms are a vital component in application architectures that use the J2EE. XML is the defined standard for asynchronous messaging and information interchange for J2EE applications. Also, XML is used by other J2EE components, such as EJBs to define deployment and initialization parameters.

XML is used throughout WebSphere Commerce Suite V5.1. It has many different uses, including store parameter definitions, instance definitions, catalog creation and description import files. This is in marked contrast to the proprietary initialization file formats used by Net.Commerce and WebSphere Commerce Suite V4.1 and the Mass Import Utility. XML provides an open industry standard of describing the WebSphere Commerce Suite system parameters and product catalog.

Java Database Connectivity (JDBC)

Java Database Connectivity, referred to as JDBC, provides a wrapper around core database functionality performed by Java applications. Within WebSphere Commerce Suite V5.1 applications, JDBC provides the underlying connectivity to the WCS instance database. JDBC technology provides an interface for the database transaction, which offers the following benefits over the database driver by providing a protective layer for the database by executing database updates in batch and connection pooling.

Within WebSphere Commerce Suite V5.1, a DataSource is defined for a JDBC connection to the database. Transactions are carried out using JDBC with the commerce-specific EJB to fulfill the business functions. This contrasts with the Net.Data model used in Net.Commerce and WebSphere Commerce Suite V4.1, where the result sets of SQL queries were processed sequentially or by using the Net.Data table functions.

In the Net.Data model, the data retrieved from the database was used directly in the display of a given page as opposed to the model-view-controller pattern used in WebSphere Commerce Suite V5.1, where the application view of the persistent data is provided by EJBs.

4.1.2 Model-view-controller (MVC) design pattern

The WebSphere Commerce Suite V5.1 application architecture uses the model-view-controller (MVC) application design pattern. The model-view-controller pattern separates display logic from the underlying business logic, and data model. The components of the model-view-controller pattern are depicted in Figure 4-1.

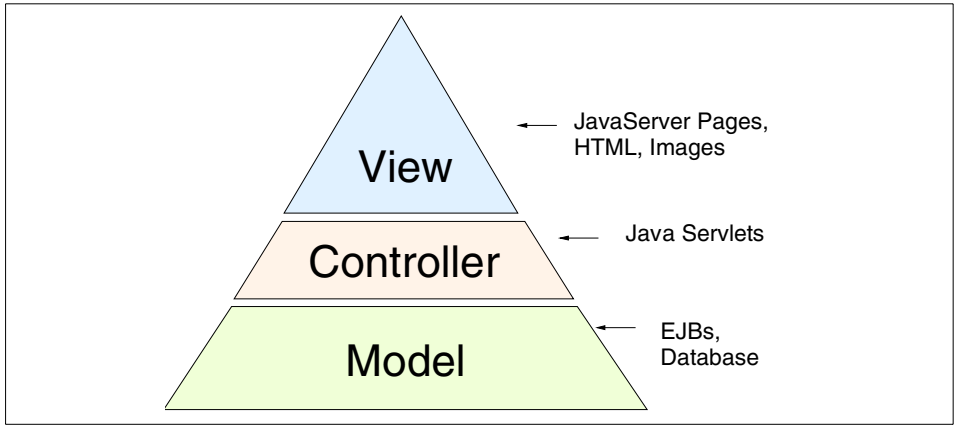


Figure 4-1 Model-view-controller design pattern

The model-view-controller (MVC) approach segregates data and business logic. In the view layer, requests are made to the controller layer for data and then displayed. The controller layer implements business logic and retrieves data from the model layer. The strength of this approach is that it delegates complex computation to components that can handle it.

Another benefit of MCV is the re-use of code. A change to the underlying structure does not necessitate a change to the view. Likewise, multiple views can interact with a single controller layer. If, for example, there were a need to extend the functionality of a site to include mobile devices, it may be necessary to develop an interface specific to the device. Using the model-view-controller development model, such a migration would only require a rewrite of the display logic, re-using the underlying business logic.

4.2 Application development overview

This section offers a high-level view of a WebSphere Commerce Suite application, and the implications that it has on development.

For more information, please refer to the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1* found on the product CD.

4.2.1 Application architecture

The WCS application architecture relationship to the model-view-controller application design pattern is depicted in Figure 4-2.

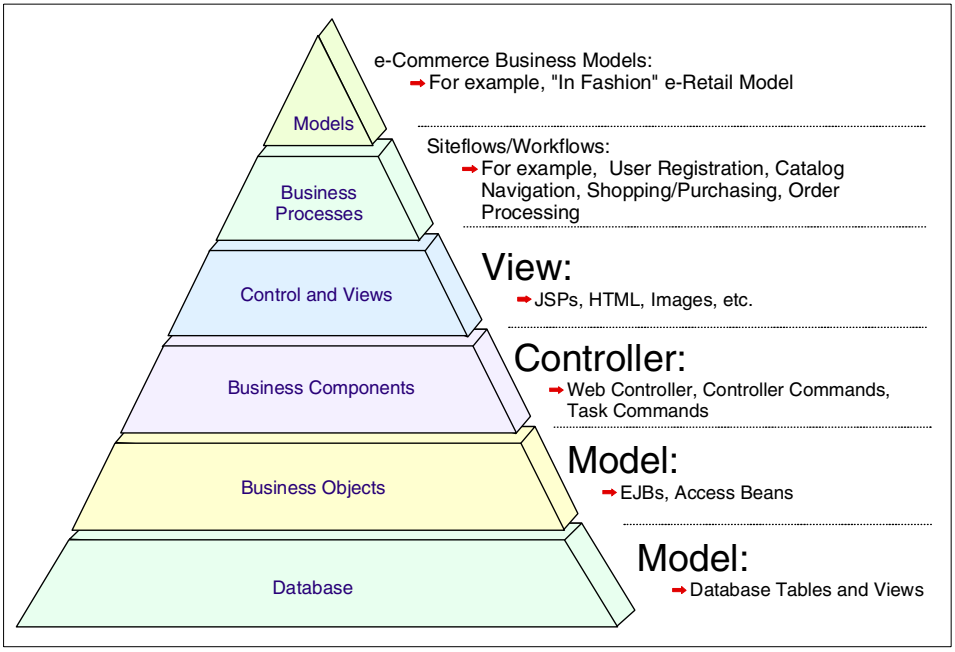


Figure 4-2 WebSphere Commerce Suite application architecture

The top two levels of this pyramid are concerned with abstract concepts of functionality in a store. They are important, but do not have a tangible form in the development model.

The highest level that actually takes the form of application code is the controls and views layer of the pyramid, which is a part of the view portion of the model-view-controller pattern.

The next level down, the business components form the controller piece of the application. Components at this level are responsible for the bulk of the computation in the application. They receive requests, perform calculations, validate input data, access the database (via access beans), etc.

Everything below the business components layer is part of the model. This layer stores the application data. It is comprised of two layers, and includes the WebSphere Commerce Suite application, business objects, and the database. Business objects are Java classes (in this application), which represent data from the database as logical objects. The database layer consists of the data itself, tables and views.

4.2.2 Approaches to development

The layered architecture above forms a complex and powerful tool for developing commerce Web sites. The architecture shields the developer from the complexity, and makes it easier to develop customizations for a commerce Web site.

The scope of a development effort for a WebSphere Commerce Suite customization depends on how far down the pyramid the customization reaches. Cosmetic changes to the view level are very simple to make. On the other hand, changes made to the database require much more effort and planning. Below we outline various levels of customization.

Basic customization

In many cases, the commerce database and WCS commands are sufficient to support the needs of a commerce Web site. All that is required is to customize the view of the site by writing custom JSPs. The architecture of the WebSphere Commerce Suite application allows for view-level customizations to be done without concern for the database structure or knowledge of the details of the command implementations.

Intermediate customization

Most customization projects fall into this category. At this level of customization, no changes need to be made to the database and the commands built into the WebSphere Commerce Suite are sufficient for most of the processing, but there is a need for some command development and custom data bean creation.

The development of a new commands often requires more effort and planning than simple cosmetic changes. Also, new commands are often accompanied by new views to represent the new logic. As such, this level of customization affects two levels of the pyramid and requires more effort.

Advanced customization

At this level of customization, extensive changes are being made to the existing application. For example, customization that affects the database schema fall into this category. If a new table needs to be added to the schema, the layers of the pyramid are affected as follows:

- Database

  The table needs to be designed, built and integrated with the existing schema.

- Business objects

  In order to support the new table, new EJBs need to be created. Data access beans need to be developed to access the new EJB(s).

- Business components

  At the controller level, the command framework must be extended to incorporate the new data.

- Controls and views

  New views need to be built to display the data to the user

Clearly, customization at this level is much more involved.

Tip: The WebSphere Commerce Suite database schema is very flexible. Often, a great deal of effort can be saved by finding a way to incorporate custom data into the existing schema.

On the other hand, if you decide to modify the WCS database, you should not make these changes lightly. We recommend that you modify the schema only if you are an advanced user.

Conforming to the WebSphere Commerce Suite model

The WebSphere Commerce Suite application architecture follows the Java server programming standard for application development. As such, it is a versatile and powerful application. During development of applications, there may be a temptation to take a short cut to circumvent the requirements of the model. For example, when adding a new table to the schema, it is theoretically possible to write commands that issue direct JDBC calls to the database, skipping the EJB and data access layers. Though this would be quicker in the short run, in the longer term it is a more costly decision. The resulting code is less modular and less maintainable. If for instance, the structure of the table changes, every command that makes use of the data would have to change as well. Also, if another application wanted to access the same data, it would need to know the table structure. It is highly recommended that customizations adhere to the existing application architecture.

4.3 Web Controller

The Web Controller is the focal point of the controller layer (in the model-view-controller) pattern for the WebSphere Commerce Server. It coordinates the activities of all of the commands in the application.

Figure 4-3 below shows the flow of a command.

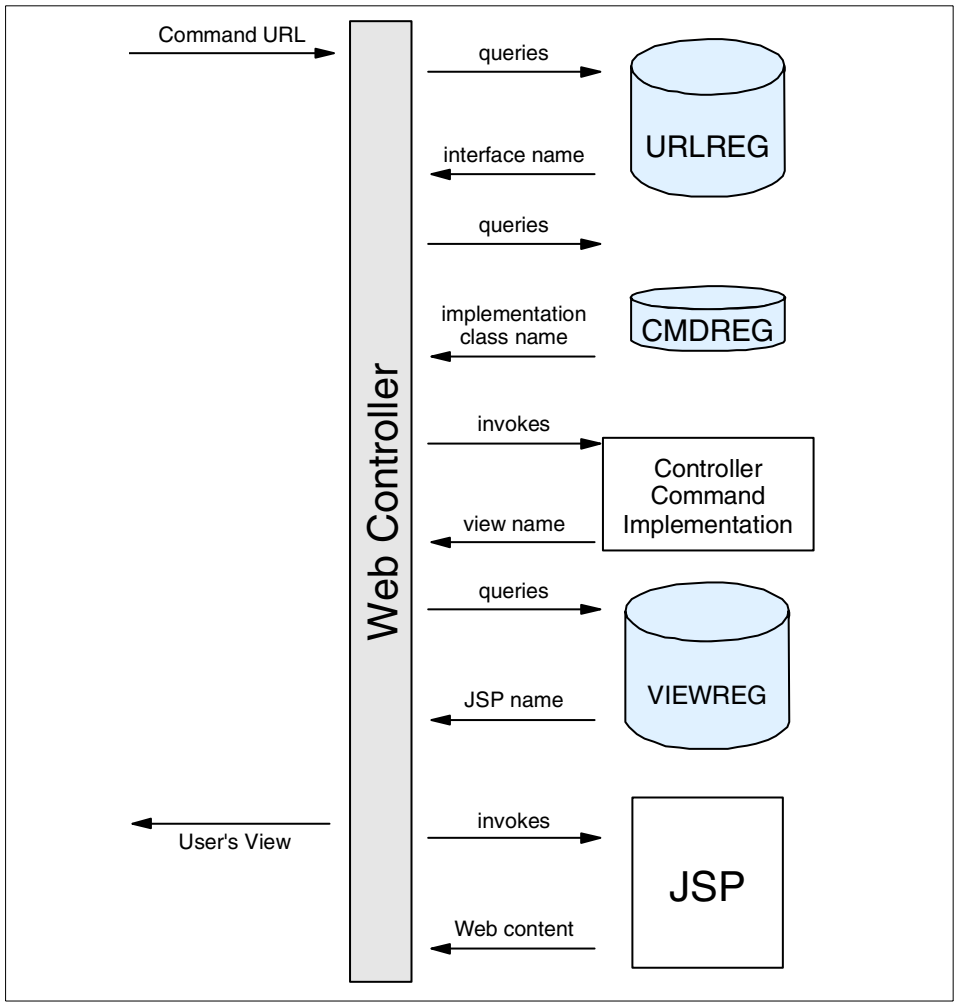


Figure 4-3 Command execution process in WebSphere Commerce Suite V5.1

When a command request URL is invoked, it is relayed to the WebSphere Commerce Suite application, where the Web Controller is its first point of contact. It coordinates the execution of the command as follows:

1. The Web Controller queries the URL registry table (URLREG) to determine which controller command interface is responsible for the requested command.
2. Having determined the controller command interface, the Web Controller maps the interface name to a controller command implementation class using the command registry table (CMDREG).

3. The Web Controller then invokes the controller command implementation.
4. The implementation class does whatever processing is required, and then returns the name of a view to the Web Controller.
5. The Web Controller maps the view name to a view command, a corresponding JSP, and any base parameters for the JSP.
6. The view command is used to invoke the JSP with the supplied parameters.
7. The JSP produces Web content.
8. The Web content is sent to the source of the request and displayed on the Web browser client.

The Web Controller handles all of the details of session control. This simplifies the work of individual commands. The details of the session are made available to the commands. The Web Controller creates and maintains a command context, which commands can query for details such as the current shopper ID, the current store ID, etc.

4.3.1 URL registry

The URL registry table URLREG is used to map URL commands to Java interface names. The key columns of the URLREG table are highlighted in Table 4-1.

Table 4-1 Key columns in the URLREG table

Database table column	Description
URI	The command issued in the URL of the request.
STOREENT_ID	The ID of the store.
INTERFACENAME	The name of the Java interface to use.
AUTHENTICATED	Indicates if user sign-in is required for this URL.
HTTPS	Indicates if secure HTTP is required for this URL.

Note: The interface used is dependent on the store for which it is invoked.

In a multiple store environment, different interfaces can be used for different stores. A value of 0 indicates all stores. This feature increases the modularity of the view level of the application.

AUTHENTICATED and HTTPS are yes/no values. They are represented by an integer: a value of 0 means no and a value of 1 indicates yes.

4.3.2 Command registry

The command registry table CMDREG works in a similar fashion to the URL registry table. It maps interface names to command implementation. In the case of the command registry, as in the URL registry, this mapping depends on the store.

The key columns of the command registry are highlighted in Table 4-2.

Table 4-2 Key columns of the CMDREG table

Database table column	Description
INTERFACENAME	The command interface.
STOREENT_ID	The ID of the store.
CLASSNAME	The Java class that will implement the interface.
PROPERTIES	The default properties associated with this command.

The class specified in the CLASSNAME column must implement (in Java) the interface specified in INTERFACENAME.

The properties column allows for specific properties to be included for the invocation of the implementation class. The properties are of the form `property1=value1&property2=value2`, where `propertyN` is the name of the property and `valueN` is the value corresponding to the property. This column allows for a single class to vary its implementation depending on the store.

4.3.3 View registry

The view registry is contained in the VIEWREG database table. It maps a view name to a view command interface and implementation. The key columns of the view registry are highlighted in Table 4-3.

Table 4-3 Key columns of the VIEWREG table

Database table column	Description
VIEWNAME	The name of the view.
STOREENT_ID	The ID of the store.
DEVICEFMT_ID	An integer specifying the device used to access the view.
INTERFACENAME	The name of the view command interface to use.
CLASSNAME	The implementation of the view command interface.

Database table column	Description
PROPERTIES	The default properties associated with this view.
HTTPS	Indicates if secure HTTP is required for this URL.

The view registry is the most difficult of the registries to understand. It is not immediately obvious where the name of the JSP is specified. It is, in fact stored in the PROPERTIES column.

The PROPERTIES column typically has the form `docname=<JSP_file_name>&property1=value1&property2=value2`. Where `propertyN` is the name of the property and `valueN` is the value corresponding to the property.

The INTERFACENAME and CLASSNAME together indicate the method to use to invoke the view. The class specified in the CLASSNAME implements a view command interface to display the JSP specified in the docname property of the PROPERTIES field, supplying additional properties as parameters to the JSP.

For more information on view commands, please refer to 4.6.1, “View commands” on page 57.

4.4 Commands

Commands are Java classes that implement the functionality of the store. Each command has a single interface and one or more implementation class. The de-coupling of the command interface and the command implementation allows multiple implementations of the same command according to context of the invocation.

In a typical command flow, a single controller command coordinates the required processing, using the command factory to invoke task commands to implement smaller pieces of logic.

4.4.1 Controller commands

A controller command is a focal point used to encapsulates the logic related to a given business process. A controller command controls the logic flow of the process using flow-control statements, such as loops and conditional statements. Also, it invokes specific task commands to implement smaller pieces of business logic.

For complex logic, controller commands may invoke one or more task commands to handle smaller subsections of logic. In this case it uses the command factory to generate the tasks command.

Figure 4-4 depicts the role of the controller command in the command flow.

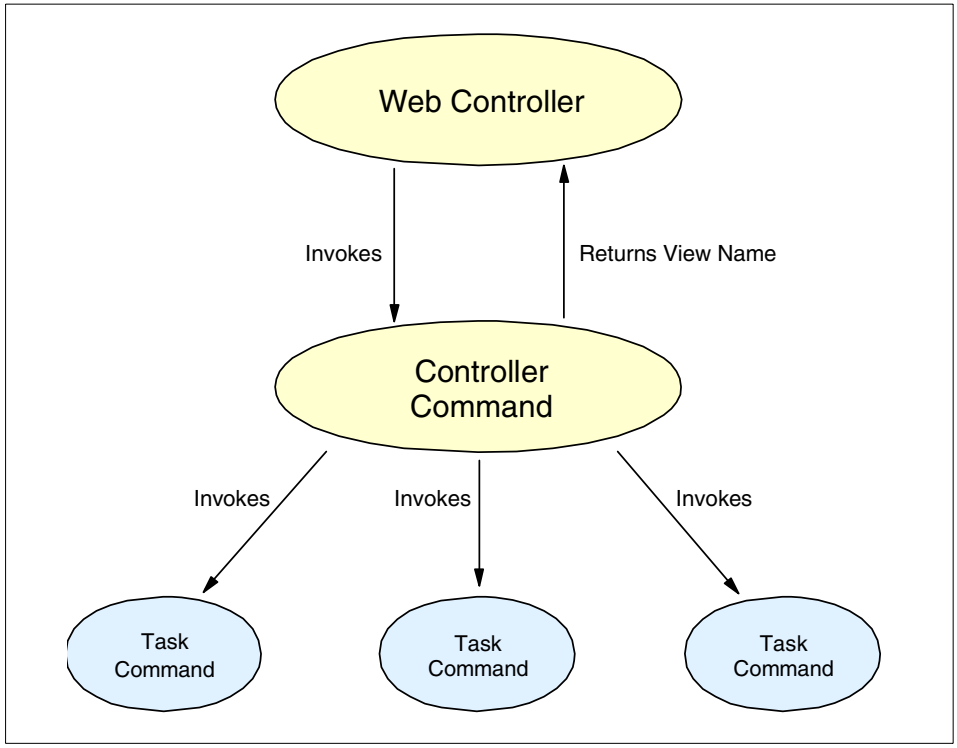


Figure 4-4 WebSphere Commerce Suite command flow

Controller commands are invoked and controlled by the Web Controller. The Web Controller uses a standard set of methods to determine which controller command should be implemented. For a detailed list of these methods, please refer to the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*.

The two key methods implemented by each controller command are as follows:

- checkParameters()

  This method is used by the Web Controller to query the command to verify that the parameters supplied are correct. The controller command should throw an exception if the parameter check fails. This exception is caught and handled by the Web Controller.

- performExecute()

  This method contains the business logic for the command. It contains the main control logic for the command. In this method, the controller command instantiates and executes task commands, and sets the view to be used upon completion.

4.4.2 Command factory

The command factory provides a mechanism for instantiating new task command objects. It is a Java class that uses the command registry to instantiate and return an implementation of the task command.

The basic syntax for using the command factory is as follows:

```
cmd= CommandFactory.createCommand(<interface_name>,commandContext.getStoreId())
```

The command factory will query the command registry to determine the correct implementation class to use for the supplied store ID. It will instantiate this class and return it to the controller command.

4.4.3 Task commands

Task commands implement smaller pieces of business logic. They are invoked by controller commands using the command factory.

Each task command should provide an implementation for the following methods:

- checkParameters()

  This method performs any parameter checking specific to this task command.

- performExecute()

  This is the method that coordinates the processing of the business logic specific to this task command. Upon its completion it should set its output variables to be read by the controller.

Each task command should provide public methods to set the input variables and public methods to get the output variables.

4.5 Data beans

Data beans are a form of command that gather data to be displayed by JSPs. They are sometimes referred to as data bean commands. In this section, we provide a simple overview of data beans. For more information, please refer to the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*.

4.5.1 Data bean manager

The data bean manager has a role similar to the command factory. It is responsible for coordinating the instantiation and population of data beans. Within a JSP, the use of the data bean manager looks as follows:

```
com.ibm.commerce.beans.DataBeanManager.activate(<data_bean>, request)
```

This call will cause the data bean manager to invoke the populate() method of the data bean.

4.5.2 Data bean implementation

Data beans are implemented by populating the data required and providing public methods to access the data.

Each data bean should have a populate method. The populate method uses the command context and the request object to coordinate its population of data. Normally, data is populated by instantiating access beans to query the EJBs, thus obtaining database data. The populate method is called by the data bean manager.

4.6 Views

A view is an important concept in the WebSphere Commerce Suite programming model. All controller commands specify a view to be invoked upon their completion. Outgoing integration messages are also implemented as views of data. A view can be seen as a combination of a view command and a JSP.

4.6.1 View commands

View commands determine the method by which the JSP is displayed. Each view is registered in the view registry (see 4.3.3, "View registry" on page 53) along with a view command interface and a view command implementation.

There are four built-in view command interfaces that developers are likely to use:

- RedirectViewCommand
- DirectViewCommand
- ForwardViewCommand
- MessagingViewCommand

The first three are the most common, while the last can be used for the construction of back-end integration messages. Also, built in are a number of common implementations of these view methods. It is highly unlikely that new view command interfaces will need to be developed and almost as unlikely that new implementations will need to be built.

For more information on this topic, refer to the WebSphere Commerce Suite online documentation, and the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*.

4.6.2 JavaServer Pages (JSPs)

JavaServer Pages or JSPs are the principle method of formatting data for display. A JSP is invoked by the view command specified in the view.

JSPs make use of data beans to display data. They carry out the view aspect of the model-view-controller design pattern. Ideally, they have a very small amount of processing and are primarily concerned with formatting the data presented by the data bean.

JSPs are also used to format outgoing XML integration messages.

4.7 Error handling and messages

Error handling is a critical element of an application. WebSphere Commerce Suite V5.1 provides a defined error handling framework for processing runtime errors and exceptions.

4.7.1 Exception handling framework

The exception handling process flow implemented in WebSphere Commerce Suite V5.1 is constructed according to the same model-view-controller pattern described in 4.1.2, "Model-view-controller (MVC) design pattern" on page 46. A command can throw one of two exceptions according to the nature of the underlying problem.

An ECApplicationException is thrown if the root cause of the exception condition is related to an action performed by the user.

An ECSystemException refers to an error resulting from a problem within the system, such as null-pointer exceptions or a database rollback condition. If the ECSystemException was caused by a database deadlock or rollback, and the command is retriable, the Web Controller will retry the command.

Error messages

Error messages are stored in stand-alone properties files, independent of the application and display logic. This is done to facilitate message maintenance and implementation of multicultural stores (multilingual). In the case of a multi-lingual store, the Web Controller determines the correct language from the specified language identifier.

The properties files define two sets of messages: user messages and system messages. User messages are messages that are presented to the application user, and system messages are captured in the system message log. All exceptions are logged by the system.

Error reporting

Error reporting is implemented by having error view tasks registered in the view registry (VIEWREG) table. When an exception is thrown by a command, it is caught by the Web Controller. The error view command corresponding to the task name defined within the exception is invoked and the display logic is processed to report the error to the user.

The JSP template is passed the same name-value pairs that were passed to the original command, along with any additional parameters added for error handling, such as an error code. In the template, an error data bean is used to present the error details in the appropriate format. A message helper object retrieves the required message from the appropriate properties file.

4.7.2 Customizing error handling and messages

For a customized store with heavily customized business logic, the error reporting will require a similar level of customization. The application developer must create three components to implement custom error messages:

- A class containing message keys.
- A class containing the message objects to be thrown and caught, and a reference to the message text resource bundle.
- A resource bundle containing the message keys and message text for each exception key.

The key names are stored independently of the exception objects to facilitate maintenance of the messages and message keys independently of the exception classes to which they refer. The resource bundle consists of key-value pairs to map the message key to the error-specific text message.

Note: All error message objects are instances of the ECMessage superclass. These objects are passed to the constructors of ECApplicationException and ECSystemExceptions when they are thrown at runtime and therefore on to the designated error display logic.

Classes using error handling, should import the com.ibm.commerce.ras package.

4.8 Debugging methods

Custom command creation will require the use of debugging tools. There are three main ways to debug customizations in WebSphere Commerce Suite:

- WebSphere Test Environment (WTE)
- WebSphere Commerce Suite log
- Program trace messages

4.8.1 WebSphere Test Environment (WTE)

The WebSphere Test Environment (WTE) is a feature of VisualAge for Java. It is a scaled-down version of the WebSphere Application Server, and allows programmers the ability to test their code without having to export the code from VisualAge for Java to the WebSphere Application Server runtime environment.

The base WCS classes are imported into VisualAge for Java and the JSPs are copied to the WTE Web path. Programmers can use VisualAge for Java's debugging tools to analyze their Java code, and the JSP Execution Monitor to analyze their JSPs.

For more information on installing the WebSphere Test Environment, please refer to the following:

- Chapter 10, "Development environment" on page 283
- *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000*, found on the product CD

4.8.2 WebSphere Commerce Suite log

WebSphere Commerce Suite maintains a running log of activity. This log serves as the standard output log as well as the standard error log. The path and name of the log file are as follows:

<wcs_install_path>/instances/<instance_name>/logs/wcs.log

It is often useful to check this log for additional debugging information. For example, JSP compilation error messages will print to this log.

4.8.3 Program trace messages

For debugging purposes, WebSphere Commerce Suite V5.1 provides a mechanism for trace logging during the execution of components running within the WebSphere Commerce Suite server.

The ECTrace class within the com.ibm.commerce.ras defines class methods for reporting entry and exit points, and runtime messages in the system trace log.

When new commands are created, tracing can be implemented as follows:

1. Determine the component that you would like to run the trace. Most often, this is the EXTERN component.
2. Create the trace entry point at the beginning of your method. This is done by inserting a statement as follows in the beginning of your method:

```
ECTrace.entry(ECTraceIdentifiers.COMPONENT_EXTERN,
      <class_name>, <method_name>)
```

3. Add trace messages as appropriate. Trace messages have the form:

```
ECTrace.trace(ECTraceIdentifiers.COMPONENT_EXTERN,
      <class_name>, <method_name>, <trace_message>)
```

4. Set the exit point for the trace as follows:

```
ECTrace.exit(ECTraceIdentifiers.COMPONENT_EXTERN,
      <class_name>, <method_name>)
```

Note: The component in which you are running the trace must be enabled in the WebSphere Commerce Suite Configuration Manager.

For more information on flow tracing, please refer to the *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*.

Part 2

Implementing the runtime environment

© Copyright IBM Corp. 2001

5

WCS runtime environment implementation overview

This chapter builds on the concepts of the runtime architecture explained in Chapter 3, “Runtime architecture” on page 29, by providing specific runtime implementation techniques and considerations for WebSphere Commerce Suite. Described are the methodologies we used for the installation of multi-tier runtimes, techniques for WCS instance creation, store deployment, and manual database schema creation.

The information is this chapter serves as a foundation for detailed instructions for implementation of the WCS runtime for the Windows NT and 2000, AIX, and Solaris platforms, in chapters to follow.

This chapter is organized into the following sections:

- Installation methodologies
- High-level installation steps: 1, 2, 3-tier
- WCS instance
- Where to find more information

© Copyright IBM Corp. 2001

5.1 Installation methodologies

This section provides an explanation of the approaches to installation based on the skill and needs of the user. As an complement to the product installation guides, we provide procedures and techniques for advanced installation scenarios.

Product installation guides

The following product installation guides (platform specific) are included on the WebSphere Commerce Suite V5.1 product CDs, which provide instructions for 1-tier and 2-tier configurations:

- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*

The product installation guides provide a sequence in which WebSphere Commerce Suite and all related products are installed automatically as part of the WebSphere Commerce Suite installation program. The guides provide information on all the possible configurable items.

Note: If you are a beginner to WebSphere Application Server and Commerce Suite, we recommend that you use the product installation guides as your primary source of information for installation and configuration.

IBM Redbook installation procedures

The installation and configuration procedures described in this chapter, and following chapters, are based on the product installation guides. We include some variations of the installation procedures to address the needs of advanced configurations. For example, we install the components individually to allow for verification steps before proceeding to the next component. In addition, we describe procedures for more advanced multi-tier configurations and techniques.

The modified installation process and technique can sometimes take longer to complete. There are several benefits of using the procedures documented in this redbook in addition to the product installation guides:

- Detailed instructions for 1, 2, 3-tier configurations (working examples)

  The product installation guides provide information on all possible values for installing an environment. We have documented procedures for installation from the approach of a working example, which includes best practices.

- Advanced topics

  We have included several advanced topics such as 3-tier and enterprise configurations, multiple instances, manual database creation and loader utility usage, and manual WCS instance deletion.

- Verification steps

  Many of the products or components that are installed build upon the successful installation of prerequisite products. We have added verifications steps to avoid the scenario of a complete installation that does not work and a user that has no knowledge of how to debug the problem.

- I/T specialist support knowledge

  If you are the user supporting a WebSphere Commerce Suite runtime environment, you will need to have a solid understanding of how the products work together to make up the complete solution. The level of understanding provided by the installation and configuration procedures provide critical information for debugging problems.

- Developer

  As a developer it is important to understand the system's architecture to have a better idea of how the store can be implemented within this environment.

Note: If you are an advanced user or someone responsible for supporting the runtime environment, you should consider using the procedures documented in this chapter as an alternative to the product installation guides for the reasons stated above.

The product installation guides are a good reference to understand all the possible values used during the installation.

5.2 High-level installation steps: 1, 2, 3-tier

The purpose of this section is to outline the high-level steps for installation and configuration for a 1, 2, and 3-tier configuration. The high-level installation steps documented are common for Windows NT, Windows 2000, AIX, and Solaris (special instructions for any platform-specific issues are noted). If you are an experienced user, these high-level steps may be sufficient to complete an installation or to be used as a guide.

Detailed instructions of the components can be found in subsequent chapters and the product installation guides.

5.2.1 High-level installation steps: 1-tier

Table 5-1 provides the high-level installation steps for the 1-tier installation.

For detailed instructions on the high-level steps outlined, refer to one of the following:

- Product installation guide for the appropriate platform
- Working example

  Chapter 6, “WCS runtime for Windows NT and 2000: DB2 and IHS” on page 87. This chapter includes detailed installation and configuration instructions in the form of a working example for a WCS V5.1 runtime on the Windows NT or 2000 platform.

Table 5-1 High-level installation steps: 1-tier

Steps	WCS Server (HTTP, DB Server, WAS, WCS)
Step 1	1. Install operating system: – Windows NT 4.0 Server + SP6a or – Windows 2000 Server + SP1 or – AIX 4.3.3 + prerequisites or – Solaris 7 + prerequisites 2. Log on as admin user – Windows NT/2000 local User Manager (not Windows Domain) or – AIX, Solaris root privilege
Step 2	Web server 1. Install Web server – IBM HTTP Server V1.3.12 or – Netscape iPlanet V4.1.5 2. Create Web server admin user 3. Configure SSL 4. Verify configuration

Steps	WCS Server (HTTP, DB Server, WAS, WCS)
Step 3	Database server 1. Install Database Server – DB2 V7.1.20 Server + Fixpak 2a or – Oracle 8i (8.1.6) 2. Update JDBC level (DB2, usejdbc2) 3. Create WAS repository database 4. Verify configuration
Step 4	WebSphere Application Server 1. Install WebSphere Application Server V3.5 AE (WAS) 2. Install WAS V3.5 Fixpak 2 (V3.5.2) 3. Apply WAS V3.5.2 E-fixes required by WCS – Copy E-fixes to \<WAS_install_path>\lib – Update \<WAS_install_path>\bin\admin.config classpath with E-fixes 4. Update Java security \<WAS_install_path>\jdk\jre\lib\security\java.security 5. Start the IBM WS AdminServer Windows service 6. Start WAS Default Server 7. Add 443 virtual host aliases 8. Verify WAS – `http://<host>/servlet/snoop` – `https://<host>/servlet/snoop` 9. If OK, stop Default Server from Admin Console (to reduce memory usage)
Step 5	WebSphere Commerce Suite 1. Install WebSphere Commerce Suite V5.1 2. Installation of WCS is complete
Step 6	Create WCS instance
Step 7	Publish store SAR file to WCS server
Step 8	Clear WAS, HTTP and WCS cache, stop/start server
Step 9	Test store

5.2.2 High-level installation steps: 2-tier

Table 5-2 provides the high-level installation steps for the 2-tier installation.

For detailed instructions on the high-level steps outlined, refer to one of the following:

- Product installation guide for the appropriate platform
- Working example

  Chapter 8, “WCS runtime for Solaris: Oracle8i and iPlanet” on page 211. This chapter includes detailed installation and configuration instructions in the form of a working example for the Solaris platform.

Table 5-2 High-level installation steps: 2-tier

Steps	Machine A (HTTP, WAS, WCS, DB client)	Machine B (DB server)
Step 1	1. Install operating system: – Windows NT 4.0 Server + SP6a or – Windows 2000 Server + SP1 or – AIX 4.3.3 + prerequisites or – Solaris 7 + prerequisites 2. Log on as admin user – Windows NT/2000 local User Manager (not Windows Domain) or – AIX, Solaris root privilege	1. Install operating system: – Windows NT 4.0 Server + SP6a or – Windows 2000 Server + SP1 or – AIX 4.3.3 + prerequisites or – Solaris 7 + prerequisites 2. Log on as admin user – Windows NT/2000 local User Manager (not Windows Domain) or – AIX, Solaris root privilege

Steps	Machine A (HTTP, WAS, WCS, DB client)	Machine B (DB server)
Step 2	Web server 1. Install Web server – IBM HTTP Server V1.3.12 or – Netscape iPlanet V4.1.5 2. Create Web server admin user 3. Configure SSL 4. Verify configuration – `http://<host>` – `https://<host>`	
Step 3		Database server 1. Install Database Server – DB2 V7.1.20 Server + Fixpack or – Oracle 8i (8.1.6) 2. Update JDBC level (DB2, usejdbc2) 3. Verify configuration
Step 4	Database client 1. Install Database Client – DB2 V7.1.20 Server + fp2a or – Oracle 8i (8.1.6) 2. Configure database client connectivity with database server 3. Create WAS repository database on remote DB server 4. Verify connectivity of database client and server	

Steps	Machine A (HTTP, WAS, WCS, DB client)	Machine B (DB server)
Step 5	WebSphere Application Server 1. Install WebSphere Application Server V3.5 AE (WAS) 2. Install WAS V3.5 Fixpak 2 (V3.5.2) 3. Apply WAS V3.5.2 E-fixes required by WCS – Copy E-fixes to \<WAS_install_path>\lib – Update \<WAS_install_path>\bin\admin.config classpath with E-fixes 4. Update Java security \<WAS_install_path>\jdk\jre\lib\security\java.security 5. Start the IBM WS AdminServer Windows service 6. Start WAS Default Server 7. Add 443 virtual host aliases 8. Verify WAS – `http://<host>/servlet/snoop` – `https://<host>/servlet/snoop` 9. If OK, stop Default Server from Admin Console (to reduce memory usage)	
Step 6	WebSphere Commerce Suite 1. Install WCS V5.1 2. Installation of WCS is complete	
Step 7	Create WCS instance	
Step 8	Publish store SAR file to WCS server	
Step 9	Clear WAS, HTTP and WCS cache, stop/start server	
Step 10	Test store	

5.2.3 High-level installation steps: 3-tier

Table 5-3 provides the high-level installation steps for the 3-tier installation.

For detailed instructions on the high-level steps outlined, refer to one of the following:

- Product installation guide for the appropriate platform
- Working example: (WAS 3-tier 1st)

  Chapter 7, “WCS runtime for AIX: DB2 and IHS” on page 137. In this working example, we configure and verify the WebSphere Application Server in a 3-tier configuration, and then install WCS. This method of building a 3-tier may be better suited for I/T specialists with a WebSphere Application Server background.
- Working example (WCS 2-tier, migrate to 3-tier)

  Chapter 6, “WCS runtime for Windows NT and 2000: DB2 and IHS” on page 87. In this working example, we build a 2-tier WCS runtime, and then migrate to a 3-tier configuration by adding a separate Web server with the WAS plug-in. This method of building a 3-tier may be better suited for I/T specialists more familiar with WebSphere Commerce Suite.

Table 5-3 High-level installation steps: 3-tier

Steps	**Machine A (HTTP Server)**	**Machine B (WAS/WCS Server)**	**Machine C (Database Server)**
Step 1	1. Install operating system and prereqs 2. Log on as admin user	1. Install operating system and prereqs 2. Log on as admin user	1. Install operating system and prereqs 2. Log on as admin user
Step 2	Web Server 1. Install Web server 2. Create Web Server admin user 3. Configure SSL 4. Verify config		

Steps	Machine A (HTTP Server)	Machine B (WAS/WCS Server)	Machine C (Database Server)
Step 3			Database Server 1. Install Database Server – DB2 V7.1.20 Server + Fixpak or – Oracle 8.1.6 2. Update JDBC level (DB2, usejdbc2) 3. Verify configuration
Step 4		Database Client 1. Install Database Client – DB2 V7.1.20 Server + Fixpack or – Oracle 8.1.6 2. Configure database client connectivity with database server 3. Create WAS repository database on remote database server 4. Verify connectivity of database client and server	

Steps	Machine A (HTTP Server)	Machine B (WAS/WCS Server)	Machine C (Database Server)
Step 5		WebSphere Application Server 1. Install WebSphere Application Server V3.5 AE (WAS) 2. Install WAS V3.5 Fixpak 2 (V3.5.2) 3. Apply WAS V3.5.2 E-fixes required by WCS – Copy E-fixes to \<WAS_install_path>\lib – Update \<WAS_install_path>\bin\admin.config classpath with E-fixes 4. Update Java security \<WAS_install_path>\jdk\jre\lib\security\java.security 5. Start the IBM WS AdminServer Windows service 6. Start WAS Default Server 7. Add 443 virtual host aliases 8. Verify WAS 9. If OK, stop Default Server from Admin Console (to reduce memory usage) 10. Confirm that static and dynamic pages are being served	

Steps	Machine A (HTTP Server)	Machine B (WAS/WCS Server)	Machine C (Database Server)
Step 6	Install WAS plug-in on Web server		
Step 7	Configure OSE remote for 3-tier, verify connectivity		
Step 8	Configure security for OSE remote IPSec		
Step 9		Install WCS V5.1	
Step 10			Manually create WCS database
Step 10		Create WCS instance	
Step 11		Publish store SAR to WCS server	
Step 12	Copy static store content to Web Server		
Step 13		Clear WAS, HTTP and WCS cache, stop/start server	
Step 14	Test store		

5.3 WCS instance

After the installation of the WebSphere Commerce Suite components, a WCS instance needs to be created.

This section is organized as follows:

- WCS instance overview
- Configuration Manager: WCS instance creation
- Command line: WCS instance creation
- Manual database creation - Loader package
- Multiple instances
- Deleting a WCS instance

5.3.1 WCS instance overview

The WCS instance is used to tie all of the pieces of the runtime together and provide an environment for the store to be deployed and run. The WCS instance includes the creation of an application server, and containers to host the WebSphere Commerce Server - <wcs_instance> within the WebSphere Application Server.

This section describes the methods of WCS instance creation and the output generated by the instance creation process.

WCS instance creation methods

There are two methods for creating a WebSphere Commerce Suite instance:

1. Configuration Manager

   WCS instances are normally created using the WCS Configuration Manager GUI using a wizard that prompts for all required information.

2. Command line

   It is also possible to create instances using the command line. This would be necessary in situations (most likely on UNIX environments, or in a server farm situation) where a local monitor is not available on the production server and the GUI cannot be used.

   In a production environment there will also be security reasons to restrict the possibilities for creating or updating instances.

WCS instance creation process and contents

The WCS instance contains the information about the following components:

- WCS instance name
- WCS instance path
- Merchant key
- Database
- Web Server
- WebSphere
- Payment Manager
- Log System
- Messaging

This section outlines the outputs of the WCS instance creation process.

- WCS instance directory structure and content

- The WCS instances file in the root of the <wcs_install_path>\instances directory contains information about the logging level and instances.
- When an instance is created, a directory is created within the WCS instance directory: <wcs_install_path>\instances\<wcs_instance>. In addition, the following subdirectories are created: cache, logs, XML.
- A WCS instance XML configuration file (<wcs_instance>.xml) can be found in the root of the WCS instance XML directory. This file contains the information about the instance and is read at startup of the instance. Changes made to this file require that the WCS Server is stopped/started. Many other XML configuration files exist in this directory.

- WAS repository database entries and WCS Server deployment

  Before creating the WCS instance, it is a requirement that the WebSphere Application Server is started. During the WCS instance creation, entries are added to the WAS database repository. In addition, the WCS Server (EJBs, Servlets, etc.) are deployed on the WebSphere Application Server.

The contents of the WCS Application Server can be seen using the WebSphere Administrators Console, displayed in Figure 5-1.

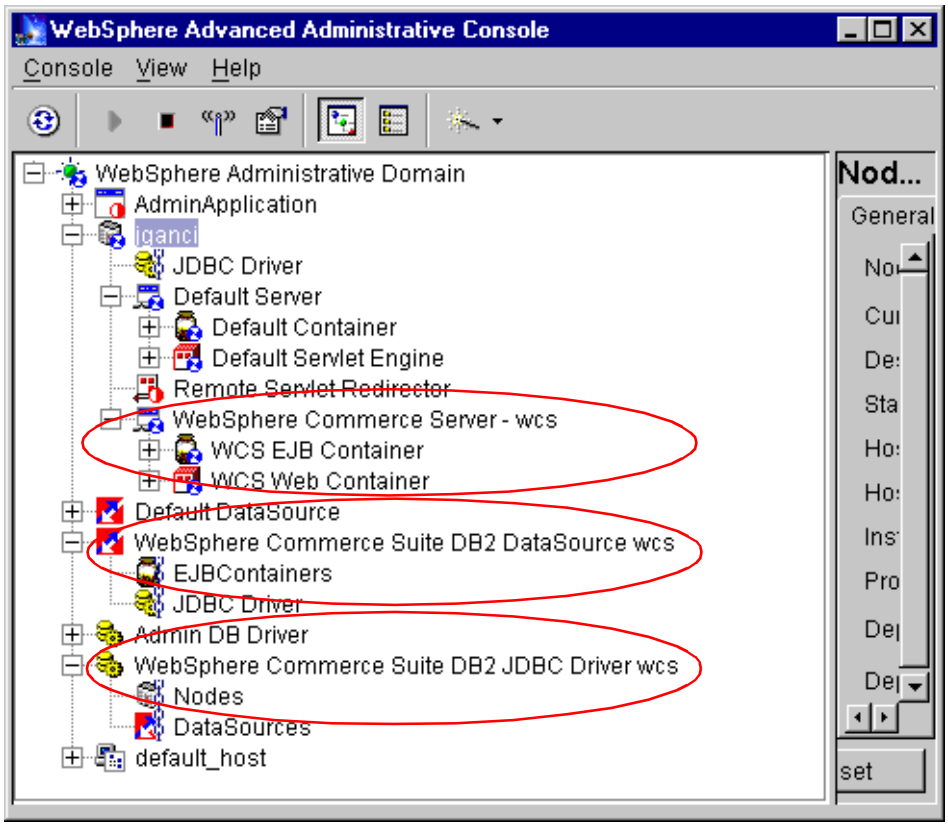


Figure 5-1 WebSphere Administrators Console - WCS instance

Important: Every time a WCS instance is created with a unique name, a WCS application server is created. This consumes a very large amount of system memory (if started).

When deleting an instance, the Configuration Manager does not remove WCS instance entries from the WAS repository database. As a result you may unintentionally be using 100+ MB of system memory for every WCS instance left in the WAS database that is not needed. You can either stop the application server for the unneeded instance or manually delete it as documented in 5.3.6, “Deleting a WCS instance” on page 84.

5.3.2 Configuration Manager: WCS instance creation

After the installation of WebSphere Commerce Suite V5.1, a WCS instance needs to be created to host the store. The Configuration Manager Java application provided in WCS V5.1 is used to create a WCS instance.

A working example of how to use the Configuration Manager to create a WCS instance, can be found in 6.6, “Create a WCS instance” on page 111.

Note: The Configuration Manager is a Java application, which requires that it be executed from the local system.

If you are operating in an environment that the WCS server does not have a console, such as a UNIX server farm, we recommend that you use the command-line method of instance creation or use a remote console.

5.3.3 Command line: WCS instance creation

The command-line method of instance creation is useful in scenarios when you do not have a console attached to the system and are working remotely from a Telnet session. Also, this method of instance creation offers the ability to provide a level of automation by using scripts to create the instance.

A working example of how to use the command line method of WCS instance creation can be found in 7.6.2, “Creating a WCS instance via the command line” on page 169.

5.3.4 Manual database creation - Loader package

Normally the WCS database and schema is created by the WebSphere Commerce Configuration Manager. There are, however, cases where you may need to create the WCS database manually before the WCS instance is created. During the WCS instance creation process, the database will be detected as already being created and continue.

There are three reasons that we encountered to create the WCS database manually:

- Database administrator (DBA)

  I/T organizations that have at least one dedicated DBA want the control and flexibility of creating the database and WCS schema on the database server. The DBA may not want to give out the DBA user ID and password to create the WCS database via the Configuration Manager. Also, the DBA may need to customize the WCS schema by adding tables, modifying tables, indexing tables, etc.

- Loader utility failures over network

  At the time of writing this redbook, we experienced network-related issues when using the WCS Configuration Manager tool, which resulted in the loader utility failing during the creation of the WCS database schema.

  To work around this issue, we created the WCS database manually on the remote database machine, prior to creating the WCS instance.

- Catalog data management

  Once your runtime environment is set up, and your store has been deployed, there is an ongoing process of maintaining the data in the WCS catalog. Products will need to be added and deleted, prices changed, etc.

  The manual database creation procedure includes manual instructions for packaging the loader utility and all required files (Java) to execute the loader on a remote system. For example, we packaged and copied the loader files from WAS/WCS server to the database server (or some other system) to use for loading the catalog.

Manual database creation procedure

This section describes the process for manually copying the loader utility and supporting files from the WCS server to the database server to create the WCS database manually on the remote Database server machine.

For a working example of the manual database creation procedure, refer to 7.8.11, “Step 11: Create the WCS database manually” on page 190.

Copy WCS loader package files from WCS server to DB server

There are a number of files and directories required to be duplicated on the database sever machine (machine C) from the WCS /WAS (machine B). This process is done to allow the loader utility to work properly to create the WCS schema. The loader utility is written Java and is dependent on the files we have documented for the WCS database schema.

Note: As an alternative to copying the loader utility directories and files from the WCS/WAS machine to the database server, install IBM Web Application Server and IBM WebSphere Commerce Suite on the database machine. Neither the WebSphere Application Server nor the Commerce Suite need to be started on this machine.

Complete the following steps to package the files necessary for the loader utility to execute on the database server:

1. Log on to the database server machine as the admin or root user. Enter a command-line window (Windows command prompt, AIX and Solaris terminal window).
2. At the command line create the directory structure on the database server machine to match the directory structure on the WAS/WCS server. For example, on AIX:

```
# mkdir /usr/lpp
# mkdir /usr/lpp/CommerceSuite
# mkdir /usr/IBMWebAS
# mkdir /usr/IBMWebAS/java
```

3. Copy the contents of the Commerce Suite instances, and the xml, bin, lib, and schema directories from the WAS/WCS machine to the Database Server machine. There is a large amount of data to copy, so ensure there is enough disk space available. For example, on AIX:

   Tar the files as follows:

```
# cd /mnt/lpp/CommerceSuite
# tar cvf /usr/lpp/CommerceSuite/instances.tar instances
# tar cvf /usr/lpp/CommerceSuite/xml.tar xml
# tar cvf /usr/lpp/CommerceSuite/lib.tar lib
# tar cvf /usr/lpp/CommerceSuite/bin.tar bin
# tar cvf /usr/lpp/CommerceSuite/schema.tar schema
# cd /mnt/IBMWebAS/java
# tar cvf /usr/IBMWebAS/was_java_lib.tar lib
# cd /mnt/IBMWebAS/java/jre
# tar cvf /usr/IBMWebAS/java/java_extra.tar lib
```

   Un-tar the files as follows:

```
# cd /usr/lpp
# mkdir CommerceSuite
# tar xvf xml.tar
# tar xvf lib.tar
# tar xvf bin.tar
# tar xvf schema.tar
# cd /usr/IBMWebAS
# tar xvf was_java_lib.tar
# cd /usr/IBMWebAS/java
# tar xvf /usr/IBMWebAS/java/java_extra.tar
```

4. For AIX and Solaris, on the database server machine, alter the permissions on the new directory structures to be 777. This is a temporary measure. For example, on AIX:

```
# chmod -R 777 /usr/lpp/CommerceSuite
# chmod -R 777 /usr/IBMWebAS/
```

Create the WCS database on the database machine

The following procedure requires that the loader utility is configured as described in the previous section. To create the WCS database manually, complete the following high-level steps:

1. On the database machine log on as root and open a terminal window.
2. Issue the following commands from the user ID that represents the instance you are creating. For example, in our case user db2inst1 on AIX, we entered the following:

   ```
   # su - db2inst1
   # cd /usr/lpp/CommerceSuite/bin
   # ./createdb.db2.sh wcsdb db2inst1 password
   ```

   A log file will be created called “creatdb.log” in the current directory. Check the contents of this file for satisfactory creation of the database.

   Note: Syntax of the createdb.db2.sh script:

   createbd2.db <database name> <instance-userid> <instance password>

3. Once the database is created, the database needs to be populated with the WCS schema. The commands of populatedb.db2.sh are as follows:

   ```
   # su - db2inst1
   # cd /usr/lpp/CommerceSuite/bin
   # ./populatedb.db2.sh wcsdb db2inst1 password
   ```

   Note: Syntax of the populatedb.db2.sh script:

   populatebd2.db <db name> <db-inst.-userid> <db-inst.passwd>

4. Run the populatedbnl.sh script from the /usr/lpp/CommerceSuite/bin directory. This script has the same syntax as the previous command:

   ```
   # populatedbnl.sh wcsdb db2inst1 password
   ```

   Once these three scripts have completed successfully, the WCS database has been created. Prior to the instance being completed on the WCS/WAS machine, the database has to be cataloged.

   Note: Syntax of the populatedbnl.db2.sh script:

   populatedbnl.db2.sh <db name> <db-inst-userid> <db-inst. passwd>

5. On the WAS machine, log on as the admin or root user and issue the following commands:

```
# su - db2inst1
# db2 connect to rs600013 user db2inst1 using password
# db2 catalog db wcsdb at node rs600013
```

6. The WCS database and schema are now complete. Next, create the WCS instance.

5.3.5 Multiple instances

There are several reasons that you may want to create more than one WCS instance on the WCS server. We have listed two possibilities:

- Multiple WCS instances same WCS database (horizontal scalability)

  This scenario is done to provide vertical scaling. By creating multiple WCS instances, you can have more simultaneous connections.

- Multiple WCS instances configured to unique WCS databases

5.3.6 Deleting a WCS instance

The Configuration Manager shipped in WCS V5.1 only partially deletes a WCS instance. For example, the entries in the WebSphere Application Server repository are not deleted. As mentioned, if the WCS instance is no longer needed and the WCS instance application server is not stopped or deleted, this will result in a very large amount of system memory being allocated to a service that is not used (100+ MB RAM).

To delete a WCS instance, complete the following steps:

1. Delete the WCS instance via Configuration Manager.
2. Manually delete or stop WCS instance application server.

   Several options will create the same end result:

   – Manually delete WCS application server entries via WebSphere Administrator Console (to reduce memory used). You will need to stop the application server, then right-click **Remove**.

     or

   – Stop the application server for the WCS instance via the Administrator's Console. This option can be used a short-term quick fix.

     or

   – If this is a test environment, or if you know how to add other customizations back to the WAS database, you can drop the WAS database, manually create it, and repopulate the WAS schema and default values (\was\bin\admin.config file, initial.install.value=true).

3. Manually delete the instance from the WCS instances directory.

5.4 Where to find more information

For additional information on configuration and scalability on the WebSphere Application Server, and WebSphere Commerce Suite, refer to the following:

- *WebSphere Commerce Suite Fundamentals Version 5.1* found on the product CD.
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition,* SG24-6153
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *WebSphere Edge Server: Working with Web Traffic Express and Net Dispatcher*, SG24-6172

6

WCS runtime for Windows NT and 2000: DB2 and IHS

This chapter includes detailed procedures for installing, configuring, verifying and deploying WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 for a 1, 2, 3-tier runtime using DB2 and IBM HTTP Server (IHS). These procedures are intended to be used as a working examples in conjunction with the product installation guides, which provide information on all the possible values that may be unique within your runtime environment.

This chapter is organized into the following sections:

- WCS runtime environment for Windows
- Install the IBM HTTP Server
- Install the DB2 Server
- Install the WebSphere Application Server
- Install the WebSphere Commerce Suite
- Create a WCS instance
- Deploy the sample store
- Installing a 2-tier runtime environment
- Installing a 3-tier runtime environment
- Where to find more information

© Copyright IBM Corp. 2001

6.1 WCS runtime environment for Windows

This section defines the hardware and software used within our test WCS runtime environment for Windows NT and Windows 2000.

6.1.1 Overview

Many of the installation and configuration steps are the same for a 1-tier, 2-tier, ad 3-tier runtime environment. The chapter is written to include a detailed procedure for installing the components for a 1-tier runtime environment. The 2-tier and 3-tier sections include the unique installation and configuration steps and refer to the common component installs of the 1-tier instructions.

The installation procedures are intended as working examples and should be used in conjunction with the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*. The value add of this chapter is the example or solution-like format and the best practices that have been integrated into the installation and configuration procedures.

1-tier runtime implementation

To implement a 1-tier environment, complete the following sections of this chapter:

- 6.1.5, “Install Windows NT/2000 and service packs” on page 90
- 6.2, “Install the IBM HTTP Server” on page 91
- 6.3, “Install the DB2 Server” on page 96
- 6.4, “Install the WebSphere Application Server” on page 100
- 6.5, “Install the WebSphere Commerce Suite” on page 108
- 6.6, “Create a WCS instance” on page 111
- 6.7, “Deploy the sample store” on page 115

2-tier runtime implementation

To implement a 2-tier environment, refer to 6.8, “Installing a 2-tier runtime environment” on page 120.

3-tier runtime implementation

To implement a 2-tier environment, refer to 6.9, “Installing a 3-tier runtime environment” on page 126.

Note: For a description of the overall flow of the installation high level steps, refer to Chapter 5, “WCS runtime environment implementation overview” on page 65.

6.1.2 Hardware and software prerequisites

For detailed information on hardware and software prerequisites, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

6.1.3 Hardware used in our test environment

This section describes the hardware used within our test WCS runtime environment for Windows NT and Windows 2000 for 1, 2, 3-tier configurations.

Note: For a 1-tier installation we recommend that new users refer to the product *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

We used several different types of systems within our test runtime environment:

- IBM xSeries Server 230 (8658)
  - 1 GHZ PIII CPU
  - 1 GB RAM
  - 18 GB Hard disk
  - 1 Ethernet (built-in)
- IBM PC 300PL (6565)
  - 733 GHZ PIII CPU
  - 512 MB RAM
  - 20 GB DASD
  - 1 IBM Etherjet 100/10
- IBM NetFinity 7000 M10 (8680)
  - 450 MHZ XEON 4-way CPU
  - 2 GB RAM
  - 36 GB DASD
  - 1 IBM Etherjet 100/10

6.1.4 Software used in our test environment

We used the following software in our test environment:

- Microsoft Windows operating systems:
  - Windows NT V4.0 Server + Service pack 6a - 128-bit encryption

    or
  - Windows 2000 Server + Service pack 1 - 128-bit encryption

> **Note:** We developed and tested the procedures on both Windows NT and Windows 2000. We did not find differences in the installation process between Windows NT and Windows 2000. In the following installation descriptions, references to “Windows” can be taken to refer to either version.

- IBM HTTP Server V1.3.12
- IBM DB2 Universal Database V7.1, Enterprise Edition for Windows + DB2 V7.1 Fixpack 2a
- IBM WebSphere Application Server V3.5, Advanced Edition + WAS V3.5 Fixpack 2 + WAS V3.5.2 E-fixes for WCS V5.1
- IBM WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 (March 2001 refresh V5.1.0.1)
- Microsoft Internet Explorer V5.5 + Service pack 1

> **Note:** Other Web servers and database software may be used, as documented in the product’s *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

6.1.5 Install Windows NT/2000 and service packs

Prior to installing the WCS runtime components, the proper level of the operating system and service pack must be installed.

1. Microsoft Windows operating systems:
   - Windows NT V4.0 Server + Service pack 6a - 128-bit encryption

     or
   - Windows 2000 Server + Service pack 1 - 128-bit encryption
2. Log on as admin user

Log on to Windows NT or Windows 2000 as a user in the local User Manager (not Windows domain).

6.2 Install the IBM HTTP Server

This section provides detailed instructions for installing, configuring, and verifying the IBM HTTP Server V1.3.12.

The IBM HTTP Server V1.3.12 installation is organized into the following phases:

- Install HTTP Server
- Configure the HTTP Server
  - Enable SSL for the HTTP Server
  - Create the HTTP Server admin user
- Verify the HTTP Server

6.2.1 Install HTTP Server

To install the IBM HTTP Server V1.3.12, complete the following steps:

1. Prior to installing the HTTP Server, we recommend that you create a Windows NT or Windows 2000 user with administrator privileges. This user ID is used by the HTTP Server to run as a Windows Service.

   For example, we created a Windows user called “httprun” with administrator group authority.
2. Insert the IBM WebSphere Application Server V3.5, Advanced Edition CD into the CD-ROM drive. This CD contains the IBM HTTP Server.
3. Using Windows Explorer, open the httpd folder on the CD, and double-click **Setup** to start the install.
4. In the Choose Setup Language window, select your national language from the drop-down menu (U.S. English is selected by default) and click **OK**.
5. Review the contents of the IBM HTTP Server 1.3.12 Installation Welcome window and click **Next**.
6. Review the contents of the Software License Agreement window, and if you accept the conditions, click **Yes** to continue.
7. In the Choose Destination Location window, click **Browse**.
8. When the Choose Folder window appears, entered the following and then click **OK**:
   - Path: `c:\ibm\http`

9. When the Setup window appears with a message `The folder does not exist, would you like to create it?`, click **Yes**.
10. Click **Next** to proceed.
11. In the Setup Type window, select **Typical** (default), and then click **Next**.
12. In the Select Program Folder window, accept the default folder (IBM HTTP Server) and click **Next**.
13. In the Information for Service Setup window, enter the following and then click **Install**:
    - User ID: `httprun`
    - Password: `<your_password>`
    - Enter the password again for verification: `<your_password>`

    Note: This is the user ID that is used by the HTTP Server Windows Service. For this example, we created a Windows user called “httprun” with administrator privileges.

    The IBM HTTP Server installation program copies files from the CD to your machine.
14. When the Setup Complete window appears, select **Yes, I will restart my computer now**, and then click **Finish**.
15. When the system restarts, log in locally as a Windows admin user and continue to the next step.

6.2.2 Configure the HTTP Server

After installing the HTTP Server, we recommend that you do the following:

- Enable SSL for the HTTP Server
- Create the HTTP Server admin user

Enable SSL for the HTTP Server

Enabling SSL for the HTTP Server is a two-step process. First, you must enable SSL by configuring the httpd.conf with SSL support. Second, you need to request and configure an SSL certificate. In this procedure we will use a self-signed test certificate.

To enable SSL for a production HTTP Server, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

Enable SSL by updating the httpd.conf

To configure SSL for the HTTP Server, complete the following steps:

1. Stop the Windows service for IBM HTTP Server and IBM HTTP Administration from Windows NT or Windows 2000 Services.
2. Use the sample httpd.conf file that includes SSL entries as a starting point to enable SSL for the IBM HTTP Server.
   a. Change to the <drive>:\<http_server_install_path>\conf directory.
   b. Back up the existing httpd.conf file by renaming it to httpd.conf.bak.
   c. Rename httpd.conf.sample to httpd.conf.
3. Modify the httpd.conf with a text editor. Ensure that the following lines are uncommented by removing the # symbol:
   - `LoadModule ibm_ssl_module modules/IBMModuleSSL<encrypt_level>.dll`

     Where `<encrypt_level>` is the appropriate level for your locale. For example, the US and Canada use 128-bit encryption.
   - `Listen 443`
   - `<VirtualHost hostname.domain.com:443>`

     You must substitute your fully qualified host name in this line.

     For example, `<VirtualHost m23bk64h.itso.ibm.com:443>`
   - `SSLEnable`
   - `</VirtualHost>`
   - `SSLDisable`
   - `Keyfile "<http_server_install_path>/keys/keyfile.kdb".`

     Change text “**keys**” to “**ssl**”.

     For example, Keyfile “c:/ibm/http/**ssl**/keyfile.kdb”
   - `SSLV2Timeout 100`
   - `SSLV3Timeout 1000`
4. Save the changes.

Create test SSL certificate

To create the self-signed test SSL certificate, complete the following steps:

1. Start the Key Management Utility by clicking **Start** -> **Programs** -> **IBM HTTP Server** -> **Start Key Management Utility**.
2. When the IBM Key Management window appears, click the **Key Database File** from the menu bar, and select **New**.
3. In the New window, enter the following and then click **OK**:

- File Name: `keyfile.kdb` (same as in httpd.conf above)
- Location: `c:\ibm\http\ssl`

4. When the Password Prompt window appears, enter the following and then click **OK**.
   - Password: `<password_http_admin_user_windows_service>`

     For example, `httprun`
   - Confirm Password: `<password_http_admin_user_windows_service>`
   - Check **Set expiration date?**, and enter number of days.

     For example, we entered `360`.
   - Check **Stash the password to a file?**
5. When the Information window appears, with a message `The password has been encrypted and saved in the file: c:\ibm\http\ssl\keyfile.sth`, click **OK**:
6. Click **Create** from the menu bar, and select **New Self-Signed Certificate** for testing.

   If you are enabling SSL production, click **New Certificate Request**.
7. When the Create New Self Signed Certificate window appears, enter the following required fields and then click **OK**:
   - Key Label: `test SSL certificate` (user defined)
   - Common Name: `<hostname.domain.com>`

     For example, `m23bk64h.itso.ral.ibm.com`
   - Organization: `IBM ITSO`
8. You should see your certificate listed. Close the IBM Key Management Utility.
9. Start the IBM HTTP Server and IBM HTTP Administration Windows services. For example, on Windows NT do the following:
   a. Click **Start** -> **Settings** -> **Control Panel** -> **Services**.
   b. Select the **IBM HTTP Server**, then click **Start**.
   c. Select the **IBM HTTP Administration**, then click **Start**.
   d. When complete, close the Windows Services window.

Create the HTTP Server admin user

To create the HTTP Server admin user, complete the following steps:

1. Open a command window, by clicking **Start** -> **Programs** -> **Command Prompt**.

2. Change to the <http_server_install_path>. For example,

   `C:\ cd \ibm\http`

3. Create the admin user by typing the following commands:

   C:\ibm\http> `htpasswd -m conf\admin.passwd httpadm`

   New password: `<your_passord>`

   Re-type new password: `<your_password>`

   Where `httpadm` is the HTTP Server Administration user ID.

4. Close the command prompt window.

6.2.3 Verify the HTTP Server

Verify that your IBM HTTP Server is operating correctly before advancing to the next WCS component installation by completing the following steps:

1. Ensure the IBM HTTP Server and IBM HTTP Administration Windows services are started.
2. Verify the non-secure HTTP Server (port 80) by entering the following URL in a Web browser:

   `http://<http_server_host_name.domain.com>`

   For example: `http://m23bk64h.itso.ral.ibm.com`

   You should see the default server home page.

3. Verify the secure HTTPS Server (port 443), by entering the following URL in a Web browser:

   `https://<http_server_host_name.domain.com>`

   For example: `https://m23bk64h.itso.ral.ibm.com`

   You should see the default server home page.

4. Verify the IBM HTTP Administration Server is working properly.
   a. Enter the following URL in a Web browser: `http://<localhost>`
   b. A Welcome to the IBM HTTP Server page is displayed; click **Configure Server**.
   c. When the Login window appears, enter the following and then click **OK**:
      - User Name: `httpadm`

        Where `httpadm` is the HTTP admin user created in the previous section.

      - Password: `<your_password>`

Note: Depending on the service level of the Microsoft Internet Explorer V5.5, you may be required to download a newer version of the JVM to run the IBM HTTP Administration Java applet.

d. Double-click **Basic Settings** from the left-hand navigation bar.

e. Double-click **Core Settings** to verify the Server name. For example, Server Name: m23bk64h.itso.ral.ibm.com.

f. Close the Web browser.

Note: Now that you have verified the server is working properly, we recommend that you stop the IBM HTTP Administration Windows service to avoid a security risk and reduce system memory consumption.

The IBM HTTP Server V1.3.12 installation is now complete.

6.3 Install the DB2 Server

This section provides detailed instructions for installing, configuring, and verifying IBM DB2 Universal Database V7.1, Enterprise Edition for Windows. The installation method described is used when installing DB2 server on a 1-tier system or on the database server of a multi-tier or remote database system.

Note: WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 includes a DB2 V7.1 CD with an internal service level of 7.1.0.24 (includes required DB2 fixes for WCS V5.1).

We recommend, however, that you install DB2 V7.1 Fixpack 2a (7.1.0.28) after installing the version of DB2 V7.1 supplied with WCS V5.1.

The DB2 V7.1 Server installation is organized into the following phases:

- Install the DB2 V7.1 Server
- Verify the DB2 V7.1 Server installation
- Install the DB2 V7.1 Fixpack 2a
- DB2 V7.1 Server configuration

6.3.1 Install the DB2 V7.1 Server

To install the DB2 V7.1 Server, complete the following steps:

1. Log in to the system where the database server will be installed using a user ID with administrator authority.
2. Insert the IBM DB2 Universal Database V7.1, Enterprise Edition for Windows CD.
3. If the install program does not start automatically, double-click **Setup** in the root directory of the CD.
4. When the Installation window appears, click **Install**.
5. When the Select Products window appears, check only the **Enterprise Edition** checkbox, and then click **Next**.
6. When the Select Installation Type window appears, select the **Typical** button, and then click **Next**.

   You may see the following message: `The OLE DB Support component is preselected by default.` Ignore the message and click **OK**.
7. When the Choose Destination Location window appears, click **Browse**.
8. When the Choose Folder window appears, enter the following and click **OK**:
   - Path: `c:\ibm\sqllib`
9. When the Setup window appears with a message `The folder does not exist, would you like to create it?`, click **Yes**.
10. Click **Next** to continue.
11. When the Enter Username and Password for Control Center Server window appears, enter the following and then click **Next**.
    - Username: `db2admin`
    - Password: `<your_password>`
    - Confirm password: `<your_password>`
    - Check the **Will allow you to use the same user ID and password throughout the installation** checkbox.
12. When you see the message `The username db2admin does not exist. Would you like Setup to create it for you?`, click **Yes**.
13. When the Start Copying Files window appears, click **Next** to start copying files.
14. When the Install OLAP Starter Kit window appears, select **Do not install OLAP Starter Kit** (it is not needed), and then click **Continue**.
15. When the Setup Complete window appears, select **Yes, I want to restart my computer now**, and then click **Finish**.

6.3.2 Verify the DB2 V7.1 Server installation

To verify the DB2 V7.1 Server installation, complete the following steps:

1. When the system restarts, log in locally as a Windows admin user.
2. Enter the required data then close the Product Registration window.
3. When the First Steps window appears, click **Exit**.
4. Open the services file located in the <drive>:\WINNT\system32\drivers\etc directory, and find the entries that have comments referring to the DB2 instance connection port. Record the service name from the first column that corresponds to the port number. For example, if the following services were displayed:

   ```
   db2cdb2 50000/tcp #Connection port for DB2 instance DB2
   db2idb2 50001/tcp #Interrupt port for DB2 instance DB2
   ```

   Record the `db2cdb2` service name and port number `50000`.
5. Verify the service name is recorded in the database manager configuration:
   a. Start the DB2 command window by clicking **Start** -> **Programs** -> **IBM DB2** -> **Command Window**.
   b. Enter the following command:

      ```
      > db2 get dbm cfg | more
      ```

   c. Look at the output for the SVCENAME entry. Verify that a value exists, and that it matches the value for the service name recorded above from the services file. For example, something similar to the following should be displayed:

      ```
      TCP/IP Service name (SVCENAME)=db2cdb2
      ```
6. If this value does not exist, update the Database Manager configuration using the following command in the DB2 command window:

   ```
   > db2 update dbm cfg using svcename <service_name>
   ```

   Where <service_name> should be replaced with the service name.

6.3.3 Install the DB2 V7.1 Fixpack 2a

To download and install DB2 V7.1 Fixpack 2a (WR21250), complete the following steps:

1. Download the DB2 UDB V7.1 Fixpack 2a from the following:

   ```
   ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2ntv7/FP2a_WR
   21250/
   ```
2. Install DB2 UDB V7.1 Fixpack 2a.

For detailed information, refer to the FixpackReadme.txt file.

6.3.4 DB2 V7.1 Server configuration

After the DB2 installation, we recommend stopping the DB2 services that are not used to conserve memory, and update the JDBC driver required by WCS.

Stop the unused DB2 services

We have listed the DB2 services that are set to *automatic* for the startup mode, after the DB2 installation. To conserve on system memory, we recommend disabling some services that are not needed by most WCS applications. The startup type for the service can be set to *automatic*, *manual*, or *disabled*. Table 6-1 lists the services settings as they are after installation, and our recommended settings.

Table 6-1 DB2 Windows services

Windows DB2 service name	Service startup type after installation	Service startup recommended setting
DB2 - DB2	automatic	automatic
DB2 - DB2CTLSV	automatic	automatic
DB2 - DB2DASD00	automatic	automatic
DB2 Governor	manual	manual
DB2 JDBC Applet Server	automatic	automatic
DB2 JDBC Applet Server - Control Center	manual	manual
DB2 License Server	automatic	automatic
DB2 Security Server	automatic	manual *
Warehouse logger	automatic	manual
Warehouse server	automatic	manual

***** We stopped the DB2 Security Server Windows service within our test environment. If your production runtime environment requires this level of security, set this service to *automatic*.

Update JDBC level

WebSphere Commerce Suite V5.1 requires using JDBC2. To update the JDBC level to JDBC2, complete the following steps:

1. Stop all DB2 services from Windows services listed in Table 6-1.

   Note: The main DB2 service will not stop until sub processes are stopped first. We stopped the services in reverse order (bottom of list to top).

2. In a command window, change to the <db2_install_path>\sqllib\java12 directory, where you installed your DB2 server and the type the following command:

   ```
   > usejdbc2
   ```

3. Restart the system.

Create the WAS database

This process creates the WebSphere Application Server repository database, also known as the WAS database. The database will be populated with a WAS schema and default values in a later step.

Alternatively, for a 1-tier configuration, you can let the createwasdb script installed by the WebSphere Application Server create the WAS database.

To create the WAS database manually, complete the following steps:

1. Open a DB2 command window by clicking **Start** -> **Programs** -> **IBM DB2** -> **Command Window**.
2. Type the following command to create the database:

   ```
   > db2 create db was
   ```

   Note: The WebSphere Application Server repository database is often called WAS. However,the database can be named any valid DB2 database name.

3. Increase the application heap size for the WAS database as follows:

   ```
   > db2 update db cfg for was using applheapsz 512
   ```

6.4 Install the WebSphere Application Server

This section describes how to install the IBM WebSphere Application Server V3.5, Advanced Edition, WebSphere Application Server V3.5 Fixpack2, and WebSphere Application Server V3.5 E-fixes required by WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000.

The WebSphere Application Server installation is organized into the following phases:

- Install WAS V3.5
- Install WAS V3.5 Fixpack 2
- Install WAS V3.5.2 E-fixes for WCS V5.1
- Configure the WebSphere Application Server
- Verify the WebSphere Application Server installation

6.4.1 Install WAS V3.5

To install the IBM WebSphere Application Server V3.5, Advanced Edition, complete the following steps:

1. Prior to the install of the WebSphere Application Server, we recommend that you create a Windows NT or Windows 2000 user with administrator privileges. This user ID is used by the WebSphere Application Server to run as a Windows Service.

   For example, we created a Windows user called “wasrun” with administrator group authority.
2. Insert the IBM WebSphere Application Server V3.5, Advanced Edition CD.
3. Start the WebSphere Application Server installation by double-clicking **Setup** from the root of the CD.
4. In the Choose Setup Language window, select your national language from the drop-down menu (English is selected by default), and click **OK**.
5. In the WebSphere Application Server Attention window, read the warnings and then click **Next** to continue.
6. In the Installation Options window, select **Custom Installation**, and then click **Next**.
7. In the Choose Application Server Components window, select/deselect the following options, and then click **Next**:
   - Select **Application and Administrative Server**
   - Select **Administrator’s Console**
   - Select **Samples** (optional)

     The samples are useful in verifying the environment.
   - Select **Web Server Plugins**
   - Select **IBM JDK 1.2.2**
   - Deselect **IBM HTTP Server** (already installed)

- Select **Configure default server and Web application**

8. In the Choose Web Server Plugins window, check **IBM HTTP Server V1.3.12**, and then click **Next**.
9. In the Security Options window, enter the username and password created with administrator privileges to allow WebSphere Application Server to run as a Windows NT or 2000 service and click **Next**.

   For example, we used wasrun (user created in step 1).
10. In the Product Directory window, enter the path to install WebSphere Application Server (for example, `c:\ibm\was` to shorten the path), and then click **Next**.

    Note: The default WebSphere Application Server path is <drive:>\WebSphere\AppServer.

    If you used the WebSphere Commerce Suite umbrella install, the WebSphere Application Server install path is \ibm\WASServer.

11. In the Database Options window, do the following:
    - Database Type: select **DB2** from the pull-down
    - Database Name: `was`

      Enter the database name for the WAS repository created on the database server.
    - Database User ID: `admin`

      By default, the Windows administrative user logged into Windows is the DB2 admin user when creating DB2 databases. In our example, the Windows administrator user ID we are logged in as is `admin`.
    - Password: `<db2_user_password>`
    - Confirm password: `<db2_user_password>`
    - Click **Next**.
12. You may receive a warning that `Your current version of DB2 does not meet level required by this product`. Click **OK** to continue. If clicking OK does not allow you to continue, click **Cancel**, and then click **Resume** to bypass the prerequisite check.

Note: At the time the IBM WebSphere Application Server V3.5, Advanced Edition was announced, DB2 V7.1 was not supported. Since the original release date, DB2 V7.1 is now supported when using Fixpack 1 or higher of the IBM WebSphere Application Server V3.5, Advanced Edition. The WebSphere Application Server install prerequisite check can be ignored.

13. In the Select Program Folder window, accept the default and click **Next** to start copying files.
14. When the Configure IBM HTTP Server window appears, specify the location of the httpd.conf file (for example, c:\ibm\http\conf directory), and then click **OK**.
15. In the Setup complete window, click **Finish**.
16. In the Restarting Windows window, select **No, I will restart my computer later**, and then click **OK**.
17. Rename createwasdb.scr in the <drive>:\<was_install_path>\bin directory to createwasdb.bak.

Note: In our example we have already created the WAS database.

18. To verify that the WebSphere Application Server has installed properly, check <drive>:\<was_install_path>\logs\wssetup.log and ensure that no errors have occurred. You should see an Install Complete statement at the end of the log to indicate a successful installation.

6.4.2 Install WAS V3.5 Fixpack 2

The IBM WebSphere Application Server V3.5, Advanced Edition Fixpack 2 is required by WCS V5.1. The fixpack is included on the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD.

To install WAS V3.5 Fixpack 2, complete the following steps:

1. Ensure the following Windows services are stopped:
   - IBM HTTP Administration
   - IBM HTTP Server
   - IBM WS AdminServer
2. Insert the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD on the Commerce Suite server machine (CD includes the WAS V3.5 Fixpack 2).

3. From a command window, change to the WebSphere Application Server Fixpack directory on the CD:

   ```
   > cd \wasfix\wasfp2
   ```

4. Copy the contents of that directory to a temporary directory on your hard drive.

   Note: Do not run the fixpak from a network drive or from the CD.

5. Change to the temporary directory you created and type `install`.
6. Enter the directory where the IBM WebSphere Application Server is installed: type `<your_was_install_path>` and then press Enter.

   For example, `c:\ibm\was`

7. Do you wish to update the WebSphere Application Server samples? (Yes/No). Type `Yes` and then press Enter.
8. Please enter the path to your Web server document root; type `<http_server_install_path>\htdocs` and then press Enter.

   For example, `c:\ibm\http\htdocs`

9. Is this correct? (Yes/No), type `Yes` if correct and then press Enter.
10. Do you wish to upgrade the IBM HTTP Server V1.3.12:(Yes/No)? Type `Yes` and then press Enter.
11. Enter the directory where the IBM HTTP Server V1.3.12 is installed. Type `<http_server_install_path>` (for example, `c:\ibm\http`).

    You should see a message stating the installation is complete.

    Confirm that the installation of WebSphere Application Server Fixpack 2 was successful by examining the <was_install_path\logs\was35_ptf_2.log file and ensuring that no errors are indicated in the file.

6.4.3 Install WAS V3.5.2 E-fixes for WCS V5.1

WCS V5.1 requires several WAS V3.5.2 E-fixes over and above the WAS V3.5 Fixpack 2 installation for proper execution. The WAS V3.5 E-fixes are included on the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD.

To install the WAS V3.5.2 E-fixes, complete the following steps:

1. Insert the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD into the CD-ROM drive of your Commerce Suite server machine.

2. Change to the <cd_drive>:\wasfix\wasefix directory.
3. Depending on your environment, copy all JAR files found in the directory's below to the <was_install_path>\lib directory:
   - WAS 3.5.2 E-fixes (required)

     For example:

     ```
     > copy f:\wasfix\wasefix\was3.5.2 efixes\*.jar c:\ibm\was\lib
     ```
   - WAS 3.5.0 E-fixes (only required if you are installing WebSphere Commerce Studio without WCS)
4. Change to the `<was_install_path>\bin` directory of the admin.config file.
5. Back up the admin.config to admin.config.bak
6. Modify admin.config file with a text editor.

   Add the following JAR files to the beginning of CLASSPATH in the admin.config file:
   - pq42952.jar
   - pq43040.jar
   - 88452_352.jar
   - 85699_352.jar
   - 88424.jar
   - 88299.jar
   - sas– r3502– 1215.jar

   Note: The original release of WCS V5.1 Pro for NT did not include the 88299.jar and included a sas-r3502-1115.jar. Make sure you update the admin.config file with the correct entries.

   For example:

   ```
   com.ibm.ejs.sm.adminserver.classpath=c:/ibm/was/lib/pq42952.jar;c:/ibm/was/
   lib/pq43040.jar;c:/ibm/was/lib/88452_352.jar;c:/ibm/was/lib/85699_352.jar;c
   :/ibm/was/lib/88424.jar;c:/ibm/was/lib/88299.jar;c:/ibm/was/lib/sas– r3502–
   1215.jar;
   ```

   Note: The above example has been split across several lines for presentation. It must actually be typed on one line.

7. Back up the modified version of admin.config to admin.config.true. This file can be used to create a clean WAS database.

Copy xmlconfig.dtd

8. Back up the <was_install_path>\bin\xmlconfig.dtd file.

   For example:

   ```
   > copy c:\ibm\was\bin\xmlconfig.dtd xmlconfig.35
   ```

9. Copy the file <cd_drive>:\wasfix\wasefix\WAS_3.5.2_efixes\xmlconfig.dtd to the <was_install_path>\bin directory (for example c:\ibm\was\bin).

Modify java.security file

10. Modify java.security file.

    a. Change to the <was_install_path>\jdk\jre\lib\security directory.

    b. Back up java.security to java.security.bak.

    c. Search for the following line in a text editor (for example, WorkPad):

    ```
    security.provider.1=sun.security.provider.Sun
    ```

    d. Add this line below it:

    ```
    security.provider.2=com.ibm.crypto.provider.IBMJCE
    ```

    Note: If you leave a trailing space at the end of the line above, the Configuration Manager will not work.

    e. Save the file.

HTTP Server fixes

11. Stop the IBM HTTP Server and IBM HTTP Administration services if started.
12. Change to the <http_install_path> and rename `ApacheCore.dll` to `ApacheCore.dll_bak`.
13. Open the <cd_drive>:\wasfix\wasefix\HTTP Server 1.3.12.1 patch directory.
14. Copy ApacheCore.dll to the HTTP Server directory (for example, c:\ibm\http).
15. Ensure the IBM HTTP Server is set to *automatic* in Windows services startup type.
16. Restart your system.

6.4.4 Configure the WebSphere Application Server

This section provides instructions for configuring the WebSphere Application Server.

1. Ensure that the DB2 services are started.
2. Ensure that the IBM HTTP Server service is started.

3. Start the WebSphere Application Server Windows service.
   a. Click **Start** -> **Settings** -> **Control Panel**.
   b. Double-click **Services**, select **IBM WS AdminServer**, and click **Start**. If desired, change the startup type to *automatic*.

   Note: The first time you start the WebSphere Application Server the default schema and configuration settings are populated. The following parameter initial.install.value=true in the admin.config file tells the application server to initialize the database. Once the server is started, the value is set to false (initial.install.value=false).

   You can recreate a clean WAS database by dropping the database, creating it, setting this initial.install.value=true, and starting the IBM WS AdminServer service.

4. Review the messages in the <was_install_path>\logs\tracefile for errors. At the end of the tracefile, you should see the following message, which means the server has started successfully:

   ```
   AdminServer   A WebSphere Administration server open for e-business
   ```

5. Start the WebSphere Application Server Administrative Console by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Application Server V3.5** -> **Administrator's Console**.
6. From the Administrator's Console, select **default_host**, and click the **Advanced** tab.
7. In the aliases field, ensure the aliases exist as seen in Table 6-2. If the aliases do not exist, add them by typing the alias in the Host Alias field, and then clicking **Apply**.

Table 6-2 WAS alias samples

Alias syntax	Alias example
<host.domain.com>	m23bk64h.itso.ral.ibm.com
<hostname>	m23bk64h
<ip_address>	9.24.105.40
<hostname.domain.com>:443	m23bk64h.itso.ral.ibm.com:443
<host>:443	m23bk64h:443
<ip_address>:443	9.24.105.40:443

6.4.5 Verify the WebSphere Application Server installation

This section provides instructions for verifying that the WebSphere Application Server has been successfully installed.

1. Ensure the DB2 services are started.
2. Ensure the IBM HTTP Server service is started.
3. Ensure the IBM WS AdminServer service is started.
4. Start the Default Server, application server, by doing the following:
   a. Start the WebSphere Application Server Administrator's Console.
   b. Select and expand **WebSphere Administrative Domain**.
   c. Select and expand your `<hostname>`.
   d. Select the **Default Server**. Right-click **Start**.
   e. When the message `Default Server start completed successfully` is displayed, click **OK**.
5. From a Web browser enter the following URLs:

   ```
   http://<hostname>/servlet/snoop
   https://<hostname>/servlet/snoop
   ```

   If snoop servlet executed correctly, the WebSphere Application Server is working properly.
6. To conserve system memory, stop the Default Server (application server) from the WebSphere Administrator's Console.

The WebSphere Application Server installation is now complete.

6.5 Install the WebSphere Commerce Suite

The WebSphere Commerce Suite installation will detect that DB2, HTTP Server, and the WebSphere Application Server are already installed and then proceed.

The WCS V5.1 installation is organized into the following phases:

- Prerequisites for WCS V5.1
- WCS V5.1 install

6.5.1 Prerequisites for WCS V5.1

Before you install WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 ensure the following prerequisites have been met:

- Prerequisite software and patches

  Prior to installing WCS V5.1, ensure that the required software components, fixpacks, and E-fixes have been installed as documented in this chapter.

- Anti-virus software

  If you are running anti-virus software you must change its startup type to *manual* in the Services window and then restart your machine before installing Commerce Suite. After you finish installing WebSphere Commerce Suite, remember to set the anti-virus service startup type back to *automatic*.

- Stop services

  Ensure that the following Windows services have been stopped:

  – IBM HTTP Server

  – IBM WS AdminServer

6.5.2 WCS V5.1 install

To install WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000, complete the following steps:

1. Prior to installing WebSphere Commerce Suite, we recommend that you create a Windows NT or Windows 2000 user with administrator privileges. This user ID is used by the WebSphere Commerce Server to run as a Windows service.

   For example, we created a Windows user called “wcsrun” with administrator group authority.

2. Insert the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD.

3. Start the WebSphere Commerce Suite installation by double-clicking **Setup** from the root of the CD.

4. When the Choose Setup Language window appears, select your national language from the drop-down menu (English is selected by default) and click **OK**.

   Note: In the Installation Options window, if your system does not meet the pre-installation requirements, a dialog box will appear detailing the requirements that have not been met. Click **Cancel** and then **Exit Setup** to exit the installation program. Take the appropriate steps to meet the pre-installation requirements.

5. When the Welcome window appears, review the contents and then click **Next**.

6. When the Software License Agreement window appears, review the agreement and then click **Accept** to continue if you accept the agreement.

   Note: If the User ID and Password window appears, enter the following and then click **Next**:

   - User ID: `wcsrun` (user defined from step 1)
   - Password: `<your_password>`

7. When the Component Selection window appears, notice the install program has detected that the IBM HTTP Server V1.3.12 and IBM Universal Database V7.1 have been installed. Click **Next**.

   Note: When installing WCS V5.1 in a 2-tier or 3-tier environment, the **Use Remote Database** checkbox must be checked.

8. When the Choose Destination window appears, click **Browse**. Enter the path to install WebSphere Commerce Suite (for example, `c:\ibm\wcs`). Click **OK**.
9. When prompted to create the directory, click **Yes**, and then click **Next**.
10. When the Select Program Folder window appears, accept the default and click **Next**.
11. When the Choose Destination Location window appears, enter the drive letter to create the installation log file, and click **Next**.
12. When the Summary window appears, review the components to be installed, and then click **Next** to start copying files.
13. When the IBM Product Registration window appears, enter the required fields or click **Exit** to register later.
14. When the Installation Complete window appears, select **Yes, I want to restart my computer now**, and then click **Finish**.
15. Verify the WCS V5.1 installation by reviewing that the wcsinstall.log file does not contain error messages. The WCSInstall.log can be found in the WCS installation root directory.

The WebSphere Commerce Suite V5.1 installation is now complete.

6.6 Create a WCS instance

After the installation of WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 and the prerequisite components, a WCS instance needs to be created. The WCS instance provides an environment for deploying and hosting a store.

There are two methods for creating a WCS instance:

- Configuration Manager - WCS instance creation

  In this chapter we describe how to create the WCS instance using the Configuration Manager, which provides a GUI.

- Command line - WCS instance creation

  For details on creating the WCS instance via the command line, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

6.6.1 Creating a WCS instance using the Configuration Manager

To create a WCS instance using the WCS Configuration Manager, complete the following steps:

1. Ensure the following Windows services are started:
   - IBM HTTP Server
   - IBM WS AdminServer
   - DB2 - DB2 (all required DB2 services)
   - IBM WCS Configuration Manager Server services
2. Start the Configuration Manager by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Configuration**.
3. When the Configuration Manager login window appears, enter the following and click **OK**:
   - User ID: `webadmin` (default)
   - Password: `webibm` (default)

   Note: The first time you log in, you will be prompted with the message `For security purposes, please change your default CM password.`

   The Configuration Manager should now be displayed.

In the remaining steps, we provide examples of our runtime environment settings. Refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000* for all the possible configuration options and values.

4. Select and expand your <hostname> from the Configuration Manager window.
5. Select **Instance List**. Right-click **Instances** -> **Create Instance**.
6. When the Select an Option window appears with the message Do you want to use an existing instance as the basis for this instance?, click **No**.

   Note: The Instance Creation Wizard window should appear. This wizard will guide you sequentially through each tab to configure components. You can go back to anyone of these tabs directly by clicking on it.

7. Select the **Instance** tab (default starting point), and enter the following and then click **Next**:
   - Instance name: wcs
   - Instance root path: c:\ibm\wcs\instances\<instance_name>

     Note: You must manually change the <instance_name> in the instance root path.

   - Merchant Key: <your_merchant_key>

     The merchant key requires a 16 character hexadecimal value. This is a user defined field.

     Note: To avoid security risk, you must enter your own defined value for the merchant key.

8. On the Database tab, enter the following and then click **Next**:
   - Database administrator name: admin
   - Database administrator password: <db_admin_password>
   - Database name: wcsdb
   - Database type: select DB2 from the pull-down
   - Database user name: admin
   - Database user password: <db_admin_password>
   - Check **Set as active database** (default)

Note: The Use remote database option must be checked if you are using a remote database (for example, 2-tier or 3-tier configuration). Refer to 6.8.7, “Step 7 (2-tier): Create a WCS instance” on page 124 for details.

9. On the Languages tab, accept the default (wcs.bootstrap_multi_en.xml), and click **Next**.
10. On the Web Server tab, enter the following and then click **Next**.
    - Hostname: `m23bk64h.itso.ral.ibm.com`

      It should detect this automatically. If not, enter the fully qualified hostname.
    - Web Server Type: select **IBM HTTP Server** from pull-down
    - Primary Document Root: `c:\ibm\http\htdocs`
    - Server Port: `80`
11. On the WebSphere tab, enter the following and then click **Next**:
    - Data Source Name: `WebSphere Commerce Suite DB2 DataSource wcs`

      The wcs instance name is added at the end of this string to create the data source name.
    - Port Number: `900` (default)
    - JDBC Driver Location: c:\ibm\sqllib\java\db2java.zip
    - Check Stores Web Application (default)
    - Check Tools Web Application (default)
12. On the Payment Manager tab, accept the defaults, and click **Next**.

    In this example, we do not configure Payment Manager.
13. On the Log System tab, accept the defaults, and click **Next**.
14. On the Messaging tab, accept the defaults, and click **Finish**.
15. When the Population Confirmation window appears, click **Yes**.

Note: The instance creation process takes approximately 10-20 minutes depending on your system specifications.

We found it helpful to view the Windows task manager CPU activity and look at the system hard disk activity LED to verify the instance creation was still in progress.

16. When the Configuration Manager is done creating the instance, you should see a message `Instance was successfully created`. Click **OK** and close the WCS Configuration Manager.

> **Note:** We recommend that you set the IBM WCS Configuration Manager Server Windows service to manual to avoid a security risk and reduce memory consumption.

17. Restart the WebSphere Commerce Server.

    After the WCS instance is created, the WebSphere Commerce Server - <wcs_instance> (application server), must be restarted by completing the following steps:

    a. Start the WebSphere Administrator's Console.

    b. Select and expand **WebSphere Administrative Domain** -> <your_host>.

    c. Select **WebSphere Commerce Server - wcs**. Right-click **Stop**. Once the server has stopped, right-click **Start**.

The WCS instance creation is now complete.

6.6.2 Verify the WCS instance

To verify the WCS instance is created properly, complete the following steps:

- Verify the existence of the instance xml configuration file <instance_name>.xml in the following directory:

  <wcs_install_path>\instances\<instance_name>\xml

  For example:

  c:\ibm\wcs\instances\wcs\xml\wcs.xml

- Review the contents of the instance creation log createdb.log in the following directory:

  <wcs_install_path>\instances\<instance_name>\logs

  For example:

  c:\ibm\wcs\instances\wcs\logs\createdb.log

- Review the contents of the wcs.log regarding the deployment of WCS within the WebSphere Application Server database repository, found in the following directory:

  <wcs_install_path>\instances\<instance_name>\logs

  For example:

c:\ibm\wcs\instances\wcs\logs\wcs.log

Use this log to ensure that the server has started correctly.

- Verify the WebSphere Commerce Suite Administration Console starts by entering the following URL in a Web browser:

  `http://<hostname>/adminconsole`

Note: The first time this tool is loaded, it takes approximately 1-2 minutes to compile the JSPs. By default, the user ID and password are wcsadmin.

6.7 Deploy the sample store

To further verify the WCS runtime for Windows, we recommend that you deploy the sample store called InFashion, supplied with WCS V5.1, using Store Services.

Note: This step is required if you intend to use this environment for development. The WebSphere Commerce Studio requires that the InFashion store be published and be named InFashion.

6.7.1 Publish store from Store Services

To deploy the sample store using Store Services, complete the following steps:

1. Start Internet Explorer V5.5 on a Windows system (browser type and level required by WCS).
2. To start Store Services, enter the following URL on Internet Explorer V5.5 Web browser: `http://<hostname>/storeservices`
3. The Security Alert window may appear. Accept the certificate and click **Yes** to continue.
4. When the Store Services Logon window appears, enter the following:
   - User name: `wcsadmin`
   - Password: `wcsadmin` (default)
   - Click **Log On**

   Note: The first time you log on, it takes a while to compile the JSPs.

5. When the Create Store Archive window appears, enter the following (see Figure 6-1):

– In the sample pull-down select **infashion_en_US_es_ES.sar**

  This sample includes multicultural support for US English, and Spanish for Spain locales. You can select any of the sample InFashion stores for your locale needs.

– Store archive (required): `InFashion`

  This is the name of the SAR file that will be generated to publish to your runtime environment. For example, InFashion.sar will be created.

– Store directory (required): `InFashion`

  This is the directory that will serve as the root for this store.

– Store owner: Default Organization (default)

– Click **OK**

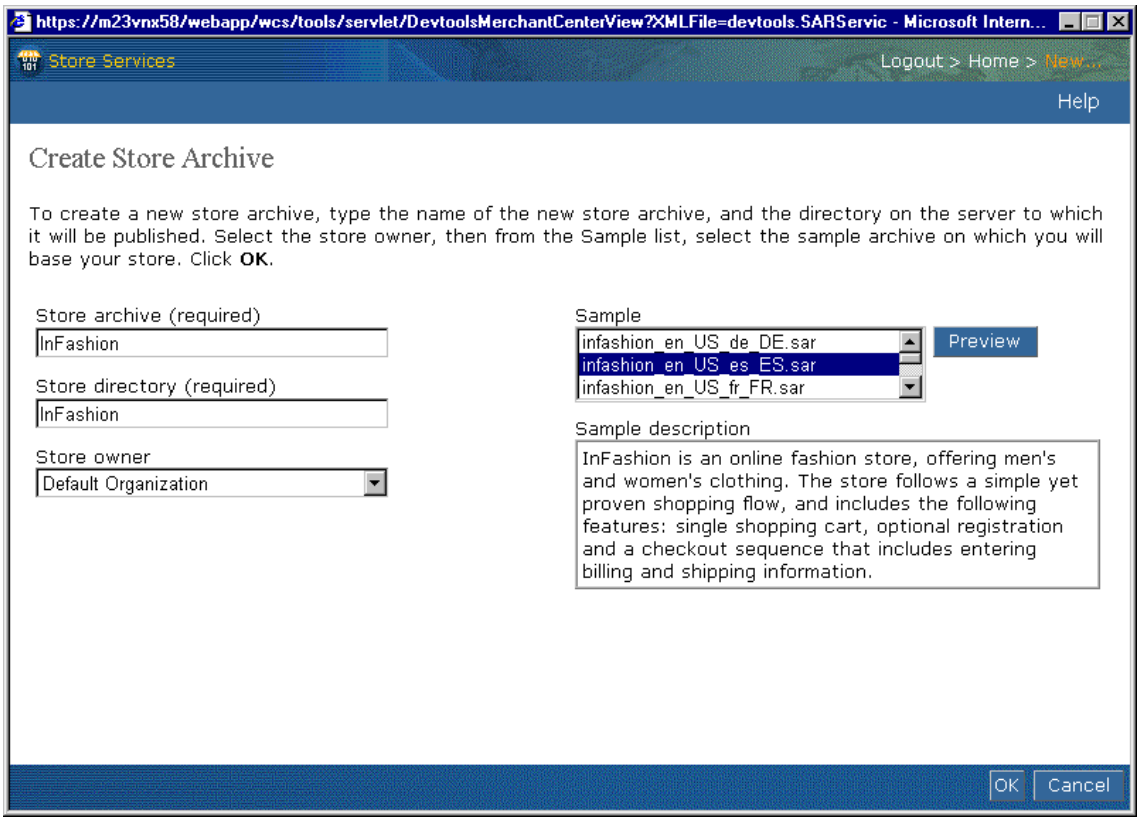


Figure 6-1 Store Services - create store archive

6. You should see a message, `InFashion.sar created successfully`. Click **OK**.

7. When the Store Archive window appears, take note of the Publish Status column of the table. The state should be Not published.
   a. Check the **InFashion.sar** checkbox under the heading Store archive.
   b. Click **Publish**.
8. When the Publish Store Archive window appears, you should see a window similar to Figure 6-2. Accept the defaults and click **OK**.

   Publishing a store take approximately 10-15 minutes.

> **Note:** We recommend viewing the Windows task manager to monitor the CPU usage, in the absence of a progress indicator for publishing the store.

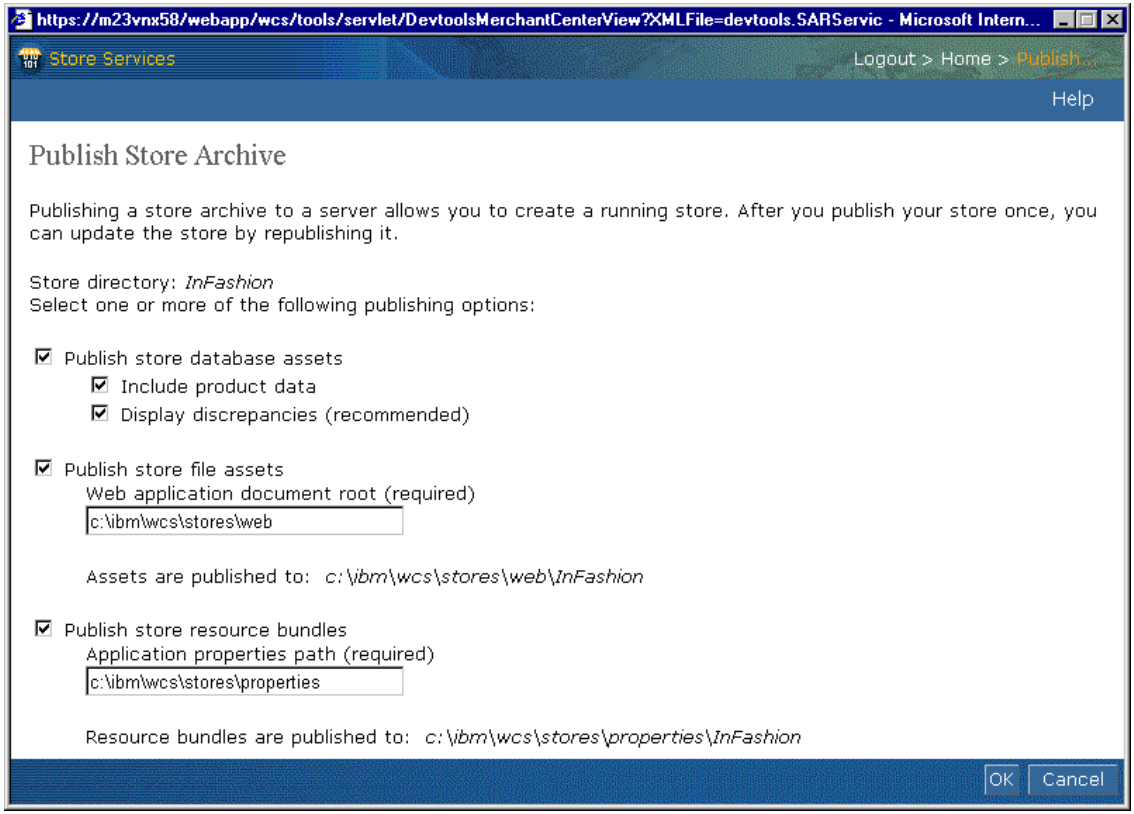


Figure 6-2 Publish store archive

9. There are several publishing status states:
   – Not published

- Awaiting publishing
- Publishing
- Published completed successfully

The first state will be Awaiting publishing. Click **Refresh**. The publishing status will change to Publishing. If successful, the status will change to Publish completed successfully.

After approximately 10 minutes, you may want to click **Refresh** to see if the publishing status has changed.

10. When the status changes to Publish completed successfully, check the **InFashion.sar** checkbox, and then click **Publish Summary**.
11. When the Publish Summary window appears, click **Launch Store** to launch the sample store.
12. You will be prompted to enter the Web application Web path. Accept the default (/webapp/wcs/stores), and click **OK**.

> **Note:** The first time the store loads it will take a while due to the compilation of store JSPs.

13. Take note of the sample store URL (bookmark or add to favorites):

```
https://<hostname>/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&langId=-1&catalogId=10001
```

> **Note:** The sample store URL launched from Store Services is an HTTPS (secure) home page. You may consider creating a bookmark to the HTTP (non-secure) Web page.

14. Close Store Services.

6.7.2 Creating an alias for the store

To make it easier for users, including yourself, to access your store, we recommend that you create an alias and an index.html file containing the location.href to your store.

Allow quick access to your store from a Web browser by typing the following URL:

- `http://<hostname>/<alias>`

  For example: `http://jganci/mystore`

Instead of:

– http://<hostname>/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&langId=-1&catalogId=10001

Create an index.html with an href for your store

To create an index.html file with the location.href for your store, complete the following steps:

1. Change to the directory of your published store. For example, c:\ibm\wcs\stores\web\.
2. Create an index.html file in this directory like the one seen in Example 6-1.
3. Change the location.href to match your store settings. For example, change the hostname.

Example 6-1 Sample index.html

```
<html>
<head>
<title>InFashion sample store</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
location.href =
"http://m23bk64h.itso.ral.ibm.com/webapp/wcs/stores/servlet/StoreCatalogDisplay
?storeId=10001&langId=-1&catalogId=10001";
</script>
</head>
<body bgcolor="#FFFFFF">
<b><font size="1" face="Arial, Helvetica, sans-serif" color="#333333">InFashion
sample store
<br>One moment.....</font></b>
</body>
</html>
```

Add the alias to the httpd.conf

To add an alias to the httpd.conf, complete the following steps:

1. Stop the IBM HTTP Server Windows service.
2. Change to the directory of the httpd.conf file (for example, c:\ibm\http\conf\).
3. Modify the httpd.conf file. Add the following alias at the end of the httpd.conf file (path of your published store):

   ```
   Alias /mystore "c:/ibm/wcs/stores/web/"
   ```

   Note: After the WCS V5.1 install, the httpd.conf file no longer is editable using Notepad (removal of crlf). We used wordpad to edit the httpd.conf.

4. Save the file.

5. Start the IBM HTTP Server service for the change to take effect.

Test the alias

Once you have updated the httpd.conf, created the index.html file, and stopped and started the IBM HTTP Server, you can now test the alias. Enter the following URL in a Web browser to access your store:

- `http://<hostname>/<alias>`

  For example: `http://m23bk64h/mystore`

6.8 Installing a 2-tier runtime environment

The two-tier system installation sequence is similar to a 1-tier system with the following exceptions:

- The database server software is installed on a separate machine.
- A DB2 client is installed on the WebSphere Commerce Server machine.
- There are additional installation steps to establish the database connection.

The following description highlights the additional installation tasks and makes references to the procedures that are common for a 1-tier configuration. To implement a 2-tier WCS environment for Windows NT or Windows 2000 using IBM HTTP Server and DB2, complete the following sections.

6.8.1 Step 1 (2-tier): Install Windows NT/2000 and service packs

Refer to 6.1.5, “Install Windows NT/2000 and service packs” on page 90.

6.8.2 Step 2 (2-tier): Install the HTTP Server

Install the IBM HTTP Server V1.3.12 on the WebSphere Commerce Suite system as detailed in 6.2, “Install the IBM HTTP Server” on page 91.

6.8.3 Step 3 (2-tier): Install the DB2 Server

Install IBM DB2 Universal Database V7.1, Enterprise Edition for Windows Server on the database server system as detailed in 6.3, “Install the DB2 Server” on page 96.

6.8.4 Step 4 (2-tier): Install the DB2 Client

The DB2 V7.1 Client is installed on the WebSphere Commerce Suite system. The DB2 V7.1 Client install is organized into the following phases:

- DB2 V7.1 Client install
- Install DB2 V7.1 Fixpack 2a
- Upgrade JDBC Level
- Configure database client/server connectivity
- Verify database client/server connectivity

DB2 V7.1 Client install

To install DB2 V7.1 Client, complete the following steps:

1. Insert the IBM DB2 Universal Database V7.1, Enterprise Edition for Windows CD into your CD-ROM drive.
2. If the DB2 installation program does not get invoked automatically, double-click **Setup** (setup.exe) file on the root of the CD.
3. In the Installation window for IBM DB2 Universal Database 7.1, click **Install**.
4. When the Select Products window, check only the **DB2 Administration Client** checkbox, and then click **Next**.
5. When the Select Installation Type window appears, select the **Typical** button, and then click **Next**.

   You may see the following message, `The OLE DB Support component is preselected by default`. Ignore the message and click **OK**.
6. When the Choose Destination Location window appears, click **Browse**.
7. When the Choose Folder window appears, enter the following and click **OK**:
   - Path: `c:\ibm\sqllib`
8. When the Setup window appears with a message `The folder does not exist, would you like to create it?`, click **Yes**.
9. Click **Next** to continue.

> **Note:** If you have configured the Windows NetBEUI protocol, DB2 will prompt you to configure NetBIOS connection for DB2. Deselect the checkbox and continue. In our example, we will configure a TCP/IP connection.

10. When the Enter Username and Password for Control Center Server window appears, enter the following and then click **Next**.

- Username: `db2admin`
- Password: `<your_password>`
- Confirm password: `<your_password>`
- Check the **Will allow you to use the same user ID and password throughout the installation** checkbox.

11. When you see the message `The username db2admin does not exist. Would you like Setup to create it for you?`, click **Yes**.
12. When the Start Copying Files window appears, click **Next** to start copying files.
13. When the Setup Complete window appears, select **No, I will restart my computer later**, and then click **Finish**.

Install DB2 V7.1 Fixpack 2a

Refer to 6.3.3, “Install the DB2 V7.1 Fixpack 2a” on page 98 for detailed instructions.

Note: At the end of the Fixpack 2a install, select **No, I will restart my computer later**, and then click **Finish**.

Upgrade JDBC Level

Refer to “Update JDBC level” on page 99 for detailed instructions.

Configure database client/server connectivity

To configure the DB2 Client, complete the following steps:

1. Open a DB2 command window on the WebSphere Commerce Server by clicking **Start** -> **Programs** -> **IBM DB2** -> **Command Window**.
2. From the command prompt, catalog the node for the remote database server:

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

   Where:

   - `<node_name>` is the unique name of your choice that DB2 will use to identify the TCP/IP node. The node name cannot be greater than eight characters.
   - `<database_server_hostname>` is the host name of your database server.
   - `<port_number>` is the port being used by DB2 (the default is 50000).

     For example:

     ```
     > db2 catalog tcpip node m23bk60l remote m23bk60l server 50000
     ```

3. Attach to the remote database node by typing the following:

```
> db2 attach to <node_name user <db2_user_ID> using <db2_userid_password>
```

Where:

- <node_name> is the unique name of your choice that DB2 will use to identify the TCP/IP node.
- <db2_user_ID> is your DB2 logon user ID.
- <db_password> is the corresponding password for <db2_user_ID>.

  For example:

  ```
  > db2 attach to m23bk60l user db2admin using db2admin
  ```

4. Catalog the remote WAS database by typing the following:

   ```
   > db2 catalog db was at node <node_name>
   ```

   Where <node_name> is the node name you used in the preceding steps.

   For example:

   ```
   > db2 catalog db was at node m23bk60l
   ```

Verify database client/server connectivity

1. To verify that the WAS database has been properly created on the remote database machine, type the following:

   ```
   > db2 list db directory
   ```

   The WAS database should have a directory entry type of Remote and a catalog node number of -1.

2. Verify the connection to the remote WAS database:

   ```
   > db2 connect to was <db2_user_ID> using <db2_userid_password>
   ```

   For example:

   ```
   > db2 connect to was user db2admin using db2admin
   ```

6.8.5 Step 5 (2-tier): Install the WebSphere Application Server

Install WebSphere Application Server on the Commerce Server as detailed in 6.4, “Install the WebSphere Application Server” on page 100.

6.8.6 Step 6 (2-tier): Install the WebSphere Commerce Suite

Install WebSphere Commerce Suite as detailed in 6.5, “Install the WebSphere Commerce Suite” on page 108 with the following exceptions.

Select a Custom Installation and when presented with the Component Selection window, and select the **Use Remote Web Server** and **Use Remote Database** options.

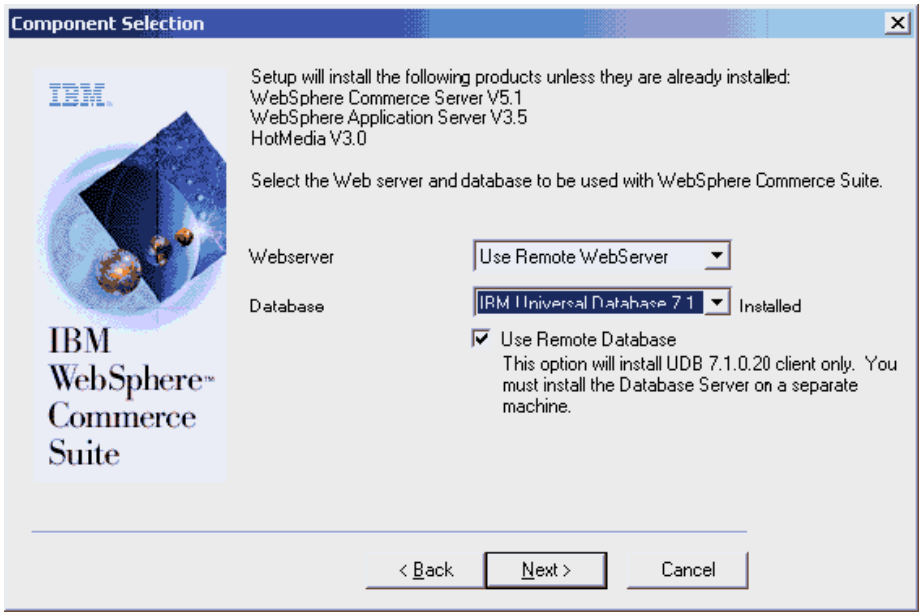


Figure 6-3 WebSphere Commerce Suite component selection

After you have installed Commerce Suite 5.1, you can create your Commerce Suite instance as described in 6.5, “Install the WebSphere Commerce Suite” on page 108. On the Web server panel of the Instance Creation wizard, select **Use Remote Webserver**.

6.8.7 Step 7 (2-tier): Create a WCS instance

The WCS instance creation for a remote database configuration (2-tier or 3-tier) is nearly identical to a 1-tier configuration (the only exception is the database configuration settings). Refer to 6.6, “Create a WCS instance” on page 111 for detailed instructions on creating a WCS instance.

When you get to the Database tab within the Configuration Manager, enter the following (see Figure 6-4) and then click **Next**:

- Database administrator name: `db2admin`
- Database administrator password: `<db_admin_password>`
- Database name: `wcsdb`

- Database type: select `DB2` from the pull-down
- Database user name: `db2admin`
- Database user password: `<db_admin_password>`
- Check **Set as active database** (default)
- Check **Use remote database**
- Database server hostname: `m23bk60l`
- Database server port: `50000`
- Database node name: `m23bk60l`

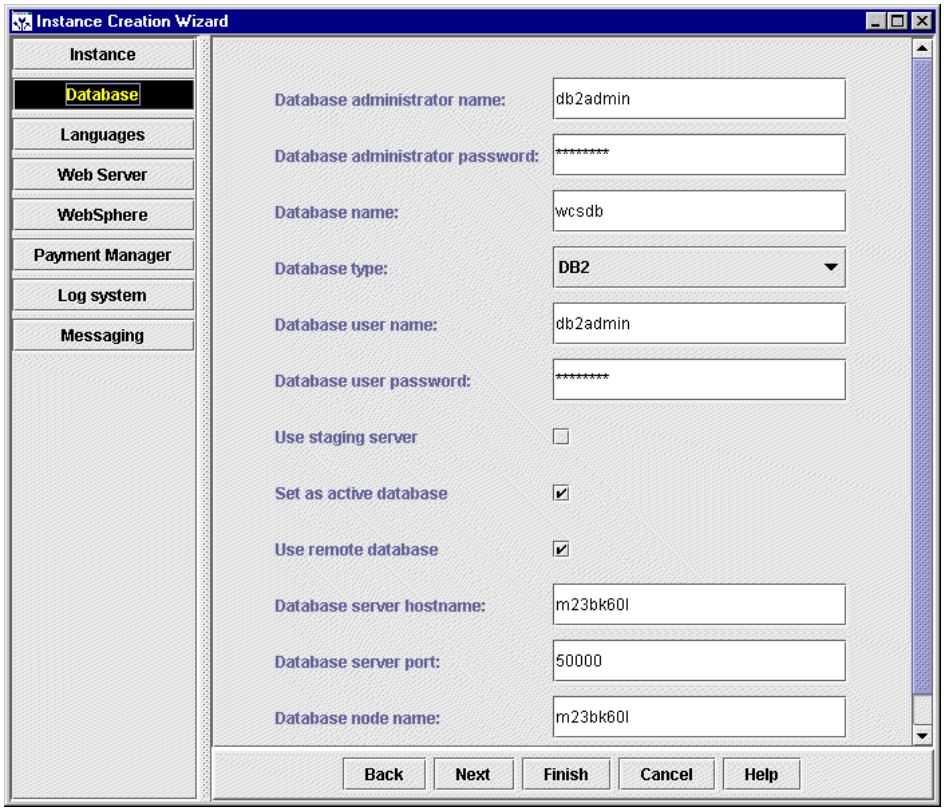


Figure 6-4 Configuration Manager: remote database

Note: You may see a message in a command window `The node name <hostname> specified in the CATALOG NODE already exists.` Ignore this message.

6.8.8 Step 8 (2-tier): Deploy the sample store

Refer to 6.7, “Deploy the sample store” on page 115 for detailed instructions.

6.9 Installing a 3-tier runtime environment

As described in 5.2.3, “High-level installation steps: 3-tier” on page 73, this 3-tier runtime environment working example starts from a 2-tier configuration. This approach to creating a 3-tier runtime may be better suited for WCS V5.1 experts. By adding a third system, Web server with the WebSphere plug-in, separate from the WebSphere Application Server, we are able to configure OSE remote to build a 3-tier WCS runtime environment.

In the new 3-tier configuration, the systems are defined as follows:

- Machine A: Web server

  For example, IBM HTTP Server and the WebSphere plug-in

- Machine B: Commerce/WebSphere Application Server

  For example, WebSphere Application Server, WebSphere Commerce Suite, and/or DB2 client.

- Machine C: Database server

  For example, DB2 server.

6.9.1 Step 1 (3-tier): 2-tier runtime configured

This configuration, relies on a 2-tier runtime already being configured. Refer to 6.8, “Installing a 2-tier runtime environment” on page 120.

6.9.2 Step 2 (3-tier): Install the HTTP Server

For example, we installed the IBM HTTP server on a separate system (machine A). Refer to 6.2, “Install the IBM HTTP Server” on page 91 for detailed instructions.

6.9.3 Step 3 (3-tier): Install the WebSphere plug-in

In order for the HTTP Server on machine A to communicate with the WebSphere Application Server on machine B, the WebSphere plug-in must be installed. The plug-in is used to provided communication between the HTTP Server and the WebSphere Application Server using the Open Servlet Engine (OSE) protocol.

WebSphere Application Server plug-in install

To install the WebSphere Application Server components required to support remote OSE, complete the following steps:

1. Ensure that the IBM HTTP Server is installed and configured with SSL support.
2. Stop the IBM HTTP Server service from Windows NT Services.
3. Start the WebSphere Application Server install program by double-clicking **Setup** from the root of the WebSphere Application Server V3.5 CD.
4. In the Choose Setup Language window, select your national language from the drop-down menu (English is selected by default) and click **OK**.
5. In the WebSphere Application Server Attention window, read the warnings and then click **Next** to continue.
6. In the Installation Options window, select **Custom Installation**, and then click **Next**.
7. In the Choose Application Server Components window, select the following options, and then click **Next**:
   - Application and Administrative Server
   - Web Server Plugins
8. In the Choose Web Server Plugins window, select IBM HTTP Server V1.3.12 and click **Next**.
9. In the Security Options window, enter the user name (for example, **“**httprun”) and the password for the IBM HTTP Server, and click **Next**.
10. In the Product Directory window, enter the path to install WebSphere Application Server (for example, c:\ibm\was) and click **Next**.
11. In the Database Options window, you can accept the default (InstantDB); InstantDB is not really used in this configuration. Click **Next**.
12. In the Select Program Folder window, accept the default and click **Next** to start copying files.
13. In the Configure IBM HTTP Server : httpd.conf window, accept the default path if it is correct, and then click **OK**. (Otherwise, modify the path to correct it as appropriate.)
14. In the Setup complete window, click **Finish**.
15. In the Restarting Windows window, select **Yes, I want to restart my computer now** and click **OK** to restart the machine.

WebSphere Application Server V3.5 Fixpack 2 install

Refer to 6.4.2, “Install WAS V3.5 Fixpack 2” on page 103 for detailed instructions.

WebSphere Application Server V3.5 E-fixes install

Refer to 6.4.3, “Install WAS V3.5.2 E-fixes for WCS V5.1” on page 104 for detailed instructions.

6.9.4 Step 4 (3-tier): Configure OSE on the WAS

To configure OSE on the WebSphere Application Server, complete the following steps:

1. Ensure that the IBM WS AdminServer service is started on machine B.
2. Launch the WebSphere Application Server Administrator’s Console.
3. Configure the WebSphere Application Server servlet engine to use INET Sockets, by completing the following steps:
   a. From the tree view in the Administrator’s Console, expand **Hostname** -> **Default Server** -> **<your_host>** -> **Default Server** -> **Default Servlet Engine**.
   b. Click the **Advanced** tab, and then click **Settings**.
   c. In the Edit Servlet Engine Transport window, select **INET Sockets** from the Transport Type drop-down list, and then click **OK**.
   d. Click **Apply**.
   e. The changes will not take effect until you stop and start the application server.
4. On Windows-based systems, change the servlet engine Maximum Connections parameter to `50` from the default value of 25.
5. Add the aliases of each Web server (machine A) to the alias list of the virtual host for the application server. In order to access WebSphere Application Server applications such as WebSphere Commerce Suite for SSL, you need to add virtual hosts. Every time you add virtual hosts you will need to rerun the OSE remote utility to regenerate the properties files for OSE remote configuration:
   a. From the tree view, select **default_host**, and then click the **Advanced** tab.
   b. Add the virtual host alias’s host name and IP address for each Web server. For ports other than port 80 (default), add the port (for example, if you are using SSL, add `<host_name>:443, x.x.x.x:443`).
   c. Press Enter after you enter each entry, and click **Apply** when you are finished to save your changes.
6. Stop and start the WebSphere Application Server, by stopping and starting the IBM WS AdminServer Windows service on machine B.

Wait for the product to refresh the automatically generated plug-in configuration files. This could take up to a few minutes.

Verify that a refresh has occurred by checking the time stamp in any of the three properties files in the temp directory under the WebSphere Application Server product installation directory. The files tell the WebSphere plug-in for the Web server how to find the remote application servers.

6.9.5 Step 5 (3-tier) Configure OSE on the HTTP Server

Now it is necessary to configure remote OSE on the Web server so it knows how to map servlet URLs to the application server machine. This is accomplished by running the remote OSE configuration script on the Web server machine to generate the necessary plug-in configuration files on the machine:

1. Open a command window and switch to the bin directory of the WebSphere Application Server installation path (for example, c:\ibm\was\bin).
2. Type the command to start and configure remote OSE:

   ```
   OSERemoteConfig -adminNodeName <AppServer_nodename>
   [-nameServiceNodeName <node_name>]
   ```

   Where:

   – <AppServer_nodename> is the WebSphere Application Server node name for the WebSphere Application Server machine to which requests should be routed.

   – <node_name> is the -nameServiceNodeName parameter, and is optional. By default, the node name and the host name are the same. If they are not, use the WebSphere Application Server name service to add the nameServiceNode parameter.

   For example:

   ```
   > OSERemoteConfig -adminNodeName m23bk64h
   ```

   The OSERemoteConfig utility generates properties files that tell the remote Web server (machine A) how to direct requests to the WebSphere Application Server (machine B). The properties files generated are located in the <was_install_path>\temp directory, and are based on the configuration data found in the WAS repository database. These files are loaded when the WebSphere Application Server bootstrap.properties file is read in by the Web server.

The properties files tell the Web server the following information:

– queues.properties

  Defines what application servers are running on which machine, and the OSE ports they are using.

– rules.properties

  Defines the Web addresses (and virtual host aliases) that correspond to a given application server.

– vhosts.properties

  Definies which IP address and host names correspond to a given WebSphere Application Server virtual host alias.

Note: OSERemoteConfig has additional, optional parameters. For details, type `OSERemoteConfig` without any parameters. Verify that the properties files contain the correct WebSphere Application Server information before proceeding.

3. Restart the IBM HTTP Server Windows services on machine A.
4. Set the IBM HTTP Server service on machine B to disable and stop the service. In our example, the HTTP Server on machine B will no longer be used.

Note: After performing any of the following activities, repeat the previous steps to rerun the OSEremote utility and restart the Web server.

- Adding a new URL (Web resource) to the environment
- Configuring a new virtual host alias
- Changing the queue properties of a servlet engine (name, port)
- Adding a new servlet engine
- Adding a new clone

5. Start the WebSphere Application Server, Default Server on machine B from the WebSphere Administrator's Console.
6. To verify that your WebSphere Application Server 3-tier configuration is working properly, enter the following URLs in a Web browser:

```
http://<webserver_hostname>/servlet/snoop
https://<webserver_hostname>/servlet/snoop
http://<webserver_hostname>/webapp/examples/ShowCfg
https://<webserver_hostname>/webapp/examples/ShowCfg
```

   Where `<webserver_hostname>` is the HTTP Server on machine A.

6.9.6 Step 6 (3-tier): Verify the default host

When using a remote Web server, the WCS Configuration Manager may define a

new virtual host when creating a WCS instance, and assign WCS servlets to use the virtual host.

The Configuration Manager checks the host name of the machine with each virtual host entry to determine if it already exists. If it finds that it does exist, the Configuration Manager will add the following default hosts:

- <hostname>
- <hostname>:443
- <hostname.domain.com>
- <hostname.domain.com>:443

If does not find that it exists, Configuration Manager will create a new virtual host, and call it `VH_<instance_name>` as seen in Figure 6-5.

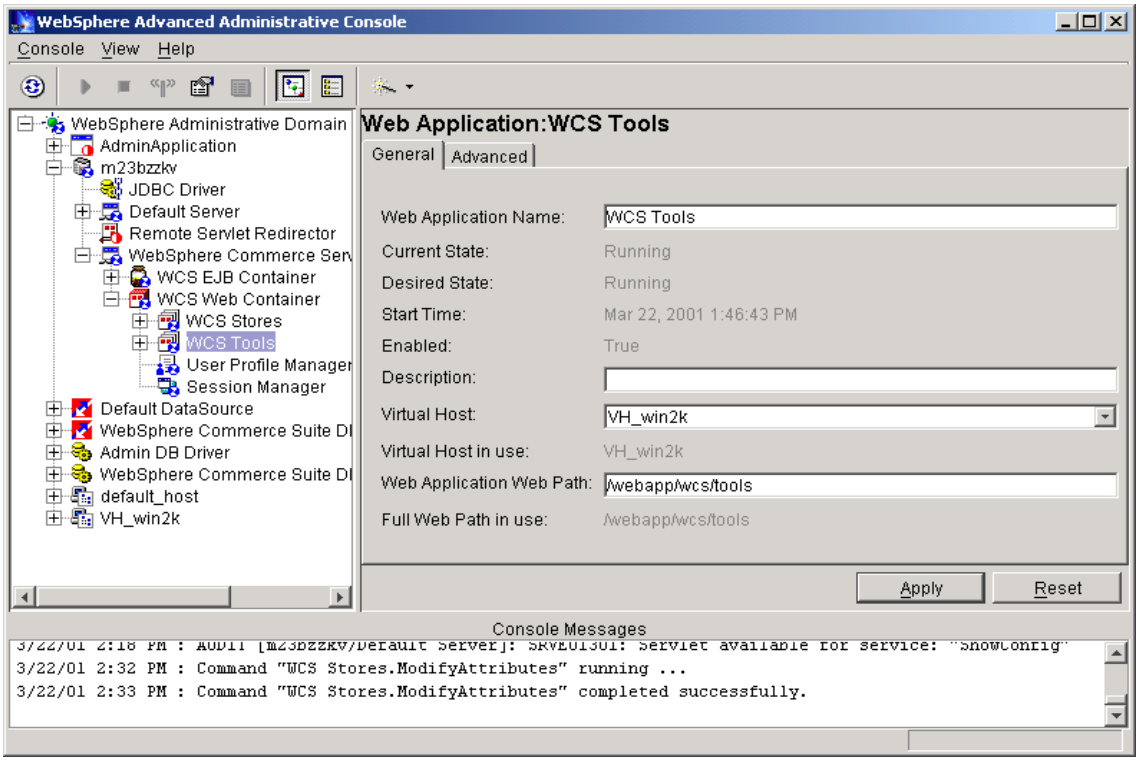


Figure 6-5 Incorrect virtual host assignment

If the servlets for a Commerce Suite instance are assigned to a different virtual host than the one you are attempting to access, your Web browser will display the following message:

```
Not Found. The requested object does not exist on this server. The link you
followed is either outdated, inaccurate or the server has been instructed
not to let you have it.
```

In this case, you need to reassign the WebSphere Commerce Server - <instance> Stores and Tools servlet groups to the default_host virtual host, by completing the following steps:

1. Start the WebSphere Application Server Administrative Console.
2. Select **WebSphere Administrative Domain** -> **<your_hostname>** -> **WebSphere Commerce Server - <instance> -> Web Container**.
3. Select **WCS Stores** (Web application). Select the **General** tab. From the Virtual Host drop-down box, select **default_host**. Click **Apply**.
4. Right-click the **VH_<instance_name>** virtual host and remove it.
5. Select **WCS Tools** (Web application). Select the **General** tab. From the Virtual Host drop-down box, select **default_host**. Click **Apply**.
6. Right-click the **VH_<instance_name>** virtual host and remove it.
7. Re-run the OSERemoteConfig utility on the Web server to reflect changes.

6.9.7 Step 7 (3-tier): Add the WCS entries to the httpd.conf

Normally, WebSphere Commerce Suite automatically updates the IBM HTTP Server httpd.conf file, when a WCS instance is created. In this example, the Web server is on a separate machine from WCS. For this reason, the WCS configuration updates need to be entered manually in the httpd.conf of the Web server (machine A).

The httpd.conf file can be found on Web server (machine A) in the <http_install_path>\conf directory. The sections listed in Example 6-2 on page 134 represent the entries needed for WCS.

1. Define the WCS VirtualHost.
   a. Insert the contents of number 1 (see Example 6-2), after `<VirtualHost` in the httpd.conf.
   b. Comment out duplicate entries from the original settings in the httpd.conf. For example:

      ```
      #<VirtualHost
      #SSLEnable
      #</VirtualHost>
      ```

   c. Replace the ServerName and IP address with your Web server settings.
2. Define the WCS file permissions.

a. Insert the contents of number 2 (see Example 6-2), after `<Directory "c:/ibm/http/cgi-bin">`.

b. Comment out duplicate entries from the original settings in the httpd.conf.

   For example:

```
#<Directory "c:/ibm/http/htdocs">
#Options Indexes
#AllowOverride None
#order allow,deny
#allow from all
#</Directory>
```

c. Change the directory as needed for your <http_install_path>.

3. Define the WCS tools aliases.

   a. Insert the contents of number 3 (see Example 6-2), after `#SetHandler server-status.`

   b. Change the paths as needed.

Example 6-2 WebSphere Commerce Suite V5.1 httpd.conf entries

```
# (1) WCS VirtualHost
# search for <VirtualHost, insert the following after it
##### IBM WebSphere Commerce Suite #####
#Instance name: wcs
<VirtualHost 9.24.105.137>
# Replace ServerName with the Web server hostname (machine A)
ServerName m23bk61w.itso.ral.ibm.com
DocumentRoot "c:/ibm/http/htdocs"
</VirtualHost>

<VirtualHost 9.24.105.137:443>
SSLEnable
SSLClientAuth 0
# Replace ServerName with the Web server hostname (machine A)
ServerName m23bk61w.itso.ral.ibm.com
DocumentRoot "c:/ibm/http/htdocs"
</VirtualHost>
#### End of IBM WebSphere Commerce Suite ####


# (2) WCS file permissions
# search <Directory "c:/ibm/http/cgi-bin">, insert the following after it
#### IBM WebSphere Commerce Suite ####
#Instance name: wcs
<Directory "c:/ibm/http/htdocs">
Options Indexes
AllowOverride None
order allow,deny
```

```
allow from all
</Directory>
<Directory "c:\ibm\wcs\web">
<Files *.jsp>
order allow,deny
deny from all
</Files>
</Directory>
#### End of IBM WebSphere Commerce Suite ####

# (3) WCS tols aliases
# search for #SetHandler server-status, insert the following after it
#### IBM WebSphere Commerce Suite ####
Alias /accelerator "c:\ibm\wcs\web\tools\common\accelerator.html"
Alias /storeservices "c:\ibm\wcs\web\tools\devtools\storeservices.html"
Alias /adminconsole "c:\ibm\wcs\web\tools\common\wcsadmincon.html"
Alias /wcs "c:\ibm\wcs\web"
Alias /wcshelp "c:\ibm\wcs\web\doc\en_US"
##### End of IBM WebSphere Commerce Suite #####
```

4. Now that all the WCS httpd.conf entries have been added, start the IBM HTTP Server Windows service.
5. Re-run the OSERemoteConfig utility on the Web server to reflect changes. For details, refer to 6.9.5, “Step 5 (3-tier) Configure OSE on the HTTP Server” on page 129.

6.9.8 Step 8 (3-tier) Serve static content from Web server

The Web server, IBM HTTP Server in this example, is more efficient at serving static content such as HTML pages and images than the WebSphere Application Server. When a store is published using WCS Store Services, all Web assets are deployed to the Commerce server system (including the WebSphere Application Server). This applies to the WCS InFashion store, and the WCS tools.

To serve static content from the remote Web server (machine A), complete the following steps:

1. WCS tools configuration.
   a. Create the c:\ibm\wcs\web directory on the Web server to match the WCS tools aliases in the httpd.conf file.
   b. Copy the contents of the c:\ibm\wcs\web directory (recursively) on the Commerce server (machine B) to the c:\ibm\wcs\web directory on the Web server (machine A).

   Note: Some of the content is optional, such as the docs directory.

c. Delete the JSPs on the Web server (machine A).

2. If your WCS store has static content, copy the static content to the directory of your alias in the httpd.conf as described in the previous step.
3. Stop/start the IBM HTTP Server Windows service on the Web server (machine A).

6.9.9 Step 9 (3-tier): Verify the sample store and WCS tools

After configuring the 3-tier runtime environment, we need to verify the InFashion sample store and WCS tools such as Store Services and the Administrator Console by completing the following steps:

1. Enter the following URL to verify the InFashion sample store:

   ```
   http://m23bk64h.itso.ral.ibm.com/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&langId=-1&catalogId=10001
   ```

2. Enter the following URLs in a Web browser and log on to verify the WCS Administrator Console and Store Services:

   ```
   http://webservername/storeservices
   http://webservername/adminconsole
   ```

Note: The WebSphere Commerce Suite Start menu items for Administrator Console and Store Services on the Commerce Server will not work, since they will point to a URL on the local machine instead of the remote Web server.

6.10 Where to find more information

- WebSphere Commerce Suite V5.1 online help
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153

7

WCS runtime for AIX: DB2 and IHS

This chapter includes detailed procedures for installing, configuring, verifying and deploying single-tier and multi-tier runtime environment for WebSphere Commerce Suite V5.1, Pro Edition for AIX using DB2 and IBM HTTP Server (IHS). These procedures are intended to be used as working examples in conjunction with the product installation guides, which provide information on all the possible values that may be unique within your runtime environment.

This chapter is organized into the following sections:

- WCS runtime environment for AIX
- Install the IBM HTTP Server
- Install the DB2 Server
- Install the WebSphere Application Server
- Install the WebSphere Commerce Suite
- Create a WCS instance
- Deploy the sample store
- Installing a 3-tier runtime environment
- Where to find more information

© Copyright IBM Corp. 2001

7.1 WCS runtime environment for AIX

This section describes the hardware and software used within our test WCS runtime environment for AIX.

Multi-tier environments offer a number of benefits over a single-tier installation. Among the benefits is the ability to provide for vertical and horizontal scaling. When implementing a 3-tier runtime environment, the separation of the HTTP Server and application server creates some minor logistical problems. One of these problems is that static pages need to be manually copied to the remote Web server, where they are served to the requesting browser client. The application logic (for example, servlets, and JSPs) needs to reside on the application server. We have documented how to work around these issues.

7.1.1 Hardware and software prerequisites

For detailed information on hardware and software prerequisites, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.

7.1.2 Hardware used in our test environment

We used the following hardware in our test environment.

WCS 1-tier runtime environment:

- 1 x IBM RS/6000 43P Model 150 with 768 MB of memory and 8 GB disk capacity

WCS 3-tier runtime environment:

- 2 x IBM RS/6000 43P Model 150 with 768 MB memory and 8 GB disk capacity
- 1 x IBM RS/6000 model F50 with 1 GB memory and 8 GB disk capacity

7.1.3 Software used in our test environment

We used the following software in out test environment:

- IBM AIX V4.3.3 + maintenance level 6
- IBM HTTP Server V1.3.12 (Apache)
- IBM DB2 UDB V7.1 EE + Fixpack 2a
- IBM Web Application Server V3.5, Advanced Edition + WAS V3.5 Fixpack 2 + WAS V3.5.2 E-fixes

- IBM WebSphere Commerce Suite V5.1, Pro Edition for AIX
- IBM JDK V1.1.8

> **Note:** The IBM JDK V1.2.2 gets installed when you install the IBM WebSphere Application Server. It does not replace the JDK V1.1.8.

The software components used within each implementation are the same. However, the functions become separated from the single machine into the multi-machine environments.

WCS 1-tier runtime environment

For a WCS 1-tier runtime environment, all components are installed on the same system, as listed:

1. Install AIX and prerequisites

   For details, refer to 7.1.4, “Install AIX” on page 140.

2. Install the IBM HTTP Server

   For details, refer to 7.2, “Install the IBM HTTP Server” on page 144.

3. Install the DB2 Server

   For details, refer to 7.3, “Install the DB2 Server” on page 151.

4. Install the WebSphere Application Server

   For details, refer to 7.4, “Install the WebSphere Application Server” on page 157.

5. Install the WebSphere Commerce Suite

   For details, refer to 7.5, “Install the WebSphere Commerce Suite” on page 164.

6. Create a WCS instance

   For details, refer to 7.6, “Create a WCS instance” on page 166.

7. Deploy the sample store

   For details, refer to 7.7, “Deploy the sample store” on page 173.

WCS 3-tier runtime environment

To implement a 3-tier WCS runtime environment, refer to 7.8, “Installing a 3-tier runtime environment” on page 179.

7.1.4 Install AIX

Prior to installing the WCS runtime environment components, the proper level of the operating system and service levels must be installed. This section provides high-level instructions for the installation of AIX for a WebSphere Commerce Suite V5.1 runtime environment. The installation of AIX is the same for multi-tier and single-tier environments.

Note: During the installation of the Web Application Server (WAS) Fixpack 2, you will need an “unzip” utility. This utility can be found at:

`http://www.info.zip.org/pub/infozip/UnZip.html#AIX.`

Once downloaded, this can be placed in the /usr/bin directory with execute permissions.

AIX 4.3.3 base operating system install

WCS V5.1 requires AIX 4.3.3 to be installed. We have listed the key things to configure during the installation and some helpful tips:

- To boot from CD for the AIX install
  - Insert AIX CD Vol 1
  - Turn on the RS/6000
  - When the keyboard POST indicator appears during startup, press F1 to enter the Systems Management Services, and set the CD as the first boot.
  - When the system reboots, it should boot from the AIX CD to start the install.
- Configure the network settings:
  - Hostname
  - IP address
  - Subnet mask
  - Default gateway IP address
  - Domain name
  - Name Server IP address
- Set the root password
- Set the paging space to at least twice the size of physical memory

Verify paging space

You must have a minimum of 128 MB of paging space. By default, the AIX configuration assistant will suggest that you create a paging file twice the size of the RAM in your system.

Complete the following steps to verify your page space.

1. Log in to AIX as root and start a terminal session..
2. Type the following command to display paging space:

   ```
   # lsps -a
   ```

   If the total active paging space is no greater than 128 MB, increase the paging space.

Increasing free space

When installing a remote database server, the pre-installation requirements will be different from a single-tier installation. Table 7-1 provides a summary of the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX* suggested values for blocks of free space for each file system.

Table 7-1 File system blocks of free space - 1-tier

File system	WCS V5.1 Install Guide - AIX suggested blocks of free space	1-tier recommended free space blocks
/ (root)	400000	400000
/home	400000	100000
/usr	4000000	4000000

We have added recommended blocks of free space for a 3-tier WCS runtime environment for AIX in Table 7-2.

Table 7-2 File system blocks of free space - 3-tier

File system	Web Server (machine A)	Commerce/WAS (machine B)	Database Server (machine C)
/ (root)	100000	400000	100000
/home	100000	100000	100000
/usr	200000	3000000	1000000

The file systems defined in Table 7-2 are described as follows:

- Increase - /

  The / (root) directory may be used for temporary files during the installation.

- Increase - /home

  By default, DB2 will create databases in the DB2 instance owner's (the instance name - in our working example, db2inst1) home directory. In the working example, we will create the WebSphere Commerce Suite (WCS) and WebSphere Application Server (WAS) repository databases on a specified file system that we will allocate storage. For this reason, we only increased /home by 100,000 blocks of free space.

- Increase - /usr

  WCS and WAS will be installed to the /usr file system on the Commerce/WAS system.

  The /usr directory is used by DB2 application files on the Database Server. In the working example, we installed the DB2 UDB Enterprise Edition with the default options, and increased /usr by 1,000,000 blocks of free space.

To increase free space, complete the following steps:

1. Before increasing the file system size, view the existing free space by using the `df` command.

   ```
   # df
   ```

2. Calculate the size required as follows:

   ```
   new_size = current_size + <desired_free_space - existing_free_space>
   ```

3. Increase the size as follows:

   ```
   # chfs -a size=<new_size> /<file_system_mount_point>
   ```

AIX 4.3.3 file sets required for WAS

There are a number of prerequisite file sets, and specific versions, required for the installation of WebSphere Application Server, and WebSphere Commerce Suite.

The AIX files sets can be installed as follows:

1. Mount the CD.
2. From the command line, type `smitty` and press Enter.
3. In the Systems Management window, select **Software Installation and Maintenance**, and press Enter.
4. In the Software Installation and Maintenance window, select **Install and Update Software**, and press Enter.
5. In the Install and Update Software window, select **Install and Update Software by Package Name**, and press Enter.

6. Insert the appropriate CD for the file sets to be added (see Table 7-3), and enter `/dev/cd0` in the INPUT field.
7. In the Select Packages to List window, select the appropriate file set to install (see Table 7-3). For example, move the cursor to **bos**, press F7 to select it, and then press Enter.
8. In the Select Software to Install window, select the desired package as seen in Table 7-3. Press Enter to install the package.

> **Note:** The "/" key may be used to search the list.

9. Repeat this process for the other file sets.

AIX prereqs for WAS V3.5

Refer to Table 7-3 for the AIX V4.3.3 file sets required by WAS V3.5.

Table 7-3 AIX prereqs for WAS V3.5

File set	Description of file set	AIX 4.3.3 CDs
bos.adt.base	Base Application Development Toolkit	CD-ROM Vol 1
bos.adt.include	Base Application Development Include Files	CD-ROM Vol 1 and 3
X11.adt.lib	AIXwindows Application Developers Toolkit Libraries	CD-ROM Vol 3
X11.adt.motif	AIXwindows Application Developers Toolkit Motif	CD-ROM Vol 3

AIX prereqs for Oracle V8.1.6

Refer to Table 7-4 for the AIX V4.3.3 file sets required by Oracle V8.1.6.

Table 7-4 AIX prereqs for Oracle V8.1.6

File set	AIX 4.3.3 CD-ROM
bos.adt.lib	CD-ROM Vol 1
bos.adt.libm	CD-ROM Vol 1

Install AIX V4.3.3 maintenance level 6

WebSphere Commerce Suite V5.1, Pro Edition for AIX requires AIX V4.3.3 maintenance level 6 or higher.

> **Note:** AIX V4.3.3 maintenance level 8 requires APAR IY19277 to fix a problem with the IBM HTTP Server.

1. AIX 4.3.3 maintenance, can be obtained from the following URL:

   ```
   http://techsupport.services.ibm.com/rs6000/fixes/
   ```

2. Install the AIX 4.3.3 maintenance level (smitty).
3. After installing AIX maintenance level 6, you can confirm the maintenance level of AIX by using the following command:

   ```
   # instfix -ik 4330-06_AIX_ML
   ```

> **Note:** After the installation of AIX service level 6, we recommend that you restart your system.

Where to find more information on AIX

The appendix on AIX in the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*, includes useful tips for common AIX tasks and commands.

In addition, the *IBM Certification Study Guide, AIX V4.3 System Administration*, SG24-5129 redbook provides an excellent source of useful AIX information.

7.2 Install the IBM HTTP Server

This section provides detailed instructions for installing the IBM HTTP Server V1.3.12 (Apache) for AIX.

The IBM HTTP Server V1.3.12 installation is organized into the following phases:

- Pre-installation tasks
- Install the HTTP Server
- Configure the HTTP Server
- Verify the HTTP Server

7.2.1 Pre-installation tasks

Prior to installing IBM HTTP Server V1.3.12, the following checks and tasks should be completed:

- Check IP ports are unused
- Check that there are no existing services on the server that use the following IP ports:
  - 80 (standard HTTP port)
  - 443 (standard HTTPS port)
  - 8008 (IBM HTTP Server Administration port)

  We suggest using the following command for this task:

```
# netstat -a | grep LISTEN
```

7.2.2 Install the HTTP Server

To install the IBM HTTP Server V1.3.12, complete the following steps:

1. Log in to AIX as user root and start an AIX terminal session.
2. Insert the WebSphere Application Server CD into the CD-ROM drive (contains the HTTP Server).
3. Mount the CD-ROM:

```
# mount -r -v cdrfs /dev/cd0 /mnt
```

4. Change the working directory to the installation path:

```
# cd /mnt/usr/sys/inst.images/ihs
```

5. Start the installation using the “smitty” tool:

```
# smitty install_all
```

6. In the Input device/directory for software field, type `./` and press Enter.
7. In the Software to Install field, press F4 to display a list of packages.
8. Scroll down the list and highlight the packages in Table 7-5, and then press F7 to select them.

Table 7-5 HTTP Server - installable packages

Package	Description
gskit	Global Security Kit (GSK) required for SSL support.
http_server.admin	Server administrator used to configure the IBM HTTP Server.

Package	Description
http_server.base	IBM HTTP Server base function.
http_server.frca	Fast Response Cache Accelerator.
http_server.html.<locale>	IBM HTTP Server documentation for the given locale. For example, the <locale> for US English is en_US.
http_server.man.en_US	IBM HTTP Server manual pages for US English (only language available).
http_server.modules **Note**: Read the description for this package	IBM HTTP Server modules. Select only the following file modules from the package: ► HTTP Server Fast-CGI ► HTTP Server MT Module If you simply select the entire package your instance will not start. The file module HTTP Server LDAP is required only when using LDAP SSL integration with the HTTP Server.
http_server.msg.<locale>.admin	IBM HTTP Server admin messages for the given <locale> (for example, en_US).
http_server.msg.<locale>.ssl.core	IBM HTTP Server SSL messages for the given <locale> (for example, en_US).
http_server.ssl.128 or http_server.ssl.56	Determines whether you will have 128-bit encryption or 56-bit encryption.
http_server.ssl.core	Required in order to install SSL modules with encryption levels.

Note: Double-check that you have selected the correct packages. Due to the number of selections, it is easy to make a mistake that will result in your HTTP Server not working properly.

Do *not* select the modules HTTP_Server.LDAP or HTTP_Server.LDAP.128 at this time. For details on LDAP support, please refer to Chapter 17, “SecureWay Directory (LDAP)” on page 471.

9. When all packages have been selected, press Enter.

10. Press the Tab key to toggle to Yes to select the following and then press Enter.
    - VERIFY install and check file sizes, Yes
    - DETAILED output, Yes
11. When the Are You Sure? message appears, press Enter.
12. When the installation is complete, the Command field at the top of the window changes from Running to OK. Scroll to the Installation Summary section at the bottom of the listing. In the Result column, you should see either SUCCESS or Already Installed next to the name of each component. If you do not, correct the problem.
13. Press F10 to return to the system prompt.
14. Unmount the CD-ROM, and remove the WebSphere Application Server CD.

7.2.3 Configure the HTTP Server

The configuration of the IBM HTTP Server V1.3.12 for AIX includes the following:

- Create the IBM HTTP Server Administrator - admin user
- Create the IBM HTTP Server - runtime user
- Configure SSL for the IBM HTTP Server
- Configure the IBM HTTP Server - ServerName

Create the IBM HTTP Server Administrator - admin user

This HTTP Server Administrator user ID and password allow access to the Administration Server Configuration GUI. To create the administration user ID, complete the following steps:

1. Change to the working directory as follows:

```
# cd /usr/HTTPServer/bin
```

2. Create the HTTP admin user (httpadm) as follows:

```
# ./htpasswd -m ../conf/admin.passwd httpadm
New password: <your_password>
Re-type new password: <your_password>
```

Note: httpasswd syntax:

```
./htpasswd -m ../conf/admin.passwd <HTTP_admin_user>
New password: <HTTP_admin_user_password>
Re-type new password:<HTTP_admin_user_password>
```

Create the IBM HTTP Server - runtime user

This procedure creates a user ID for the HTTP Server to run after the user root has started the server process. To create the HTTP Server runtime user, group, and permissions, complete the following steps:

1. Change to the working directory as follows:

   ```
   # cd /usr/HTTPServer/bin
   ```

2. Execute the following script included with the IBM HTTP Server:

   ```
   # ./setupadm
   ```

3. Answer the prompts as follows:

   a. Please supply a user ID to run the Administration Server. For example, enter `httprun` and then press Enter.

   b. Please Supply a Group Name to run the Administration Server. For example, enter `httpgrp` and then press Enter.

   c. Please supply the directory containing the files for which a change in the permissions is necessary. The default is /usr/HTTPServer/conf. Press Enter to accept the default.

   d. Please supply the file name for permission changes. Default will change the file permissions for all files in the /usr/HTTPServer/conf directory. Press Enter to accept the default.

   e. Special permissions are required for the Administration Server to handle the restart of managed HTTP Server. This will update the group, group access permissions, and group execute permissions for the /usr/HTTPServer/bin/admrestart file. This interface is necessary for the Administration server to manage the HTTP Server.

      Enter `1` (yes) and then press Enter.

   f. The configuration file /usr/HTTPServer/conf/admin.conf will be saved. Do you wish to update the Administration Server Configuration file? To perform the change, enter 1. To exit with no change, enter 2 (default).

      Enter `1` to update configuration file and then press Enter.

   g. Do you wish to run the IBM HTTP Administration Server and IBM HTTP Server in a language other than English? For a language other than English, enter 1. For English, enter 2 (default).

      Press Enter to accept the default (English).

The setupadm program should return to a system prompt.

Configure SSL for the IBM HTTP Server

Enabling SSL support for the IBM HTTP Server is required for WCS V5.1. To configure the IBM HTTP Sever with SSL support, complete the following steps:

1. Stop the IBM HTTP Server:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl stop
   ```

2. Copy the sample HTTPd.conf:

   ```
   # cd /usr/HTTPServer/conf
   # cp httpd.conf httpd.conf.orig
   # cp httpd.conf.sample httpd.conf
   ```

3. Modify the httpd.conf with a text editor (for example, vi). Uncomment the following lines by removing the # sign:

   ```
   #LoadModule ibm_ssl_module libexec/mod_ibm_ssl_<encryption_level>.so
   ```

   Where the `<encryption_level>` is the appropriate level for your locale (for example, 128 for en_US).

   Remove the comment character from the following lines to configure the SSL module in the HTTP Server.

   ```
   #AddModule mod_ibm_ssl.c
   #Listen 443
   #<VirtualHost host.domain.com:443>
   ```

   Note: You must substitute your fully qualified host name in this line (for example, `<VirtualHost jganci2.itso.ral.ibm.com:443>`.

   Uncomment the `SSLEnable` section:

   ```
   #SSLEnable
   #</VirtualHost>
   #SSLDisable
   #Keyfile "/usr/HTTPServer/keys/keyfile.kdb"
   ```

   Note: Replace the word **keys** with **ssl**.

   Uncomment the SSL timeout values:

   ```
   #SSLV2Timeout 100
   #SSLV3Timeout 1000
   ```

4. Save your changes.

5. Set the Java path by typing the following:

   ```
   # JAVA_HOME="/usr/jdk_base"
   # export JAVA_HOME
   ```

6. Start IBM Key Management:

   ```
   # ikeyman
   ```

7. When the IBM Key Management window appears, click the **Key Database File** menu and select **New.**
8. When the New window appears, enter the following and then click **OK**:
   - File name: `keyfile.kdb`
   - File location: `/usr/HTTPServer/ssl`
9. When the Password Prompt window appears, enter the following and then click **OK**:
   - IBM HTTP Server runtime password: `<your_password>`

     This is the password for the runtime user created in “Create the IBM HTTP Server - runtime user” on page 148.
   - Check **set expiration time** and change it to the desired number of days (for example, 360).
   - Enable **Stash the password to a file.**
10. When the Information window appears, confirming the password has been saved and encrypted in the keyfile.sth file, click **OK.**

> **Note:** For production SSL enablement, refer to the SSL enablement section in the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.
>
> For the purposes of testing, we created a self-signed certificate.

11. Click **Create** and select **New Self-Signed Certificate.**
12. When the Create New Self-Signed Certificate window appears, enter the following (required fields) and then click **OK**.
    - Key Label: `ITSO test SSL certificate` (user-defined name)
    - Organization: `IBM` (your company name)
13. Close the Key Management Utility.
14. Start HTTP Server as follows:

```
# cd /usr/HTTPServer/bin
# ./apachectl start
```

15. Start IBM HTTP Administration Server as follows:

```
# cd /usr/HTTPServer/bin
# ./adminctl start
```

Configure the IBM HTTP Server - ServerName

To configure the IBM HTTP Server, ServerName stored in the httpd.conf, complete the following steps:

1. Enter the following URL in a Web browser: `http://<hostname>`
2. When the Welcome to the IBM HTTP Server window appears, click **Configure server**.
3. When the IBM HTTP Administration Server login window appears, enter the following and then click **OK**:
   - User ID: `httpadm`
   - Password: `<your_password>`
4. Double-click **Basic Settings** from the left-hand navigation bar.
5. Double-click **Core Settings** and enter the server host name:
   - Server Name: `<fully_qualified_hostname>`
   - Click **Submit** to save (scroll down).
6. Close the Web browser.

7.2.4 Verify the HTTP Server

To verify the HTTP Server is working properly, complete the following steps:

1. Enter the following URL in a Web browser to verify the HTTP Server:

   `http:\\<your_hostname>`

2. Enter the following URL in a Web browser to verify that SSL is enabled for the HTTP Server:

   `https:\\<your_hostname>`

3. Close the Web browser.

The IBM HTTP Server V1.3.12 for AIX installation is now complete.

7.3 Install the DB2 Server

This section provides detailed instructions for installing the IBM DB2 Universal Database V7.1, Enterprise Edition for AIX Server. The instructions can be used to install the DB2 Server in a single-tier or multi-tier configuration. Included in the instructions are best practices in preparation for the install by creating file systems for the databases.

The DB2 Server installation is organized into the following phases:

- Pre-installation requirements
- Install the DB2 V7.1 Server
- Install the DB2 V7.1 Fixpack 2a
- Create the WAS database on the DB2 Server

Note: When installing and configuring DB2 V7.1 EE for AIX in a remote database configuration (2-tier, 3-tier), there are some special instructions that are needed in addition to the procedure documented in this section. These additional instructions are documented in 7.8, “Installing a 3-tier runtime environment” on page 179.

WebSphere Commerce Suite V5.1, Pro Edition for AIX includes a CD for DB2 UDB V7.1.0.20, EE for AIX. After installing DB2 V7.1, we recommend that you install DB2 V7.1 Fixpack 2a (V7.1.0.28).

7.3.1 Pre-installation requirements

This section provides step-by-step instructions for preparing your system for a DB2 Server installation, specific to WebSphere Commerce Suite V5.1.

Create dedicated storage for the DB2 instance

The default database location for DB2 is the instance owner’s home directory. In practice many users will create a file system to store their databases. Using a file system with access permissions provides the instance owner with control and the ability to set the storage size of the file system. In the working example we create a logical volume and file system to store the remote databases for the WebSphere Application Server and the WebSphere Commerce Server.

Create a logical volume

The following are instructions for creating a logical volume on AIX:

1. Log in to AIX as root and start a terminal session.
2. Start smitty as follows:

   ```
   # smitty storage
   ```

3. Select **Logical Volume Manager** and press Enter.
4. Select **Logical Volumes** and press Enter.
5. Select **Add a Logical Volume** and press Enter.
6. Press F4 in the VOLUME Group name: field to display a list.
7. Select **rootvg** and press Enter.

8. When a Logical Volume window appears, enter the following fields:
   - Logical Volume NAME: `dblv1`
   - Number of LOGICAL PARTITIONS: `1`
9. Press Enter. When complete, the status should change to Command: OK.
10. Press the F10 key to return to the system prompt.

Create a journal file system - mount point

Create a file system with the mount point to the home directory of the DB2 instance owner. The DB2 instance owner will be created in subsequent steps.

1. Start smitty as follows:

   ```
   # smitty storage
   ```

2. Select **File Systems**.
3. Select **Add / Change / Show / Delete File Systems**.
4. Select **Journal File System**.
5. Select **Add a Journaled File System on a Previously Defined Logical Volume**.
6. Select **Add a Standard Journal File System**.
7. When the Add a Standard Journal File System window appears, do the following:
   - LOGICAL VOLUME name: press F4 for listing, select **dblv1**
   - MOUNT POINT: `/home/<db2_instance_owner>`

     For example, `/home/db2inst1`
   - Mount AUTOMATICALLY at system restart: select **yes**

Note: Press the Tab key to toggle yes/no.

   Press Enter. When complete, the status should change to Command: OK.
8. Press the F10 key to return to the system prompt.

Allocate storage

If you have not restarted your system, it is necessary to mount the file system prior to allocating storage to the newly created file system.

1. Mount file system:

   ```
   # mount /dev/dblv1 /home/db2inst1
   ```

2. Allocate storage

   This step will allocate file system storage space to the journal file system. By default, AIX allocates storage in 512-byte blocks. In this example we will create a file system that is 800,000 blocks or 400 MB. As the WCS database and WAS database grow, this file system may need to be increased.

   ```
   # chfs -a size='600000' /home/db2inst1
   ```

3. Verify file system free space:

   ```
   # df
   ```

Create groups and users

The DB2 installation has the capability to automatically create a user and groups to manage a DB2 instance. To have better control over the instance, we choose to manually create groups, and users and make the db2inst1 user the owner of the /home/db2inst1 file system (mount point) that has 400 MB of dedicated storage. During the DB2 Server installation, we will create a DB2 instance and use the user db2inst1, and home directory /home/db2inst1. We recommend that you use the same name for the user and DB2 instance name.

To create DB2 groups and users, complete the following steps:

1. Log in to AIX as root and start a terminal session..
2. Create group db2iadm1 by typing the following:

   ```
   # mkgroup db2iadm1
   ```

3. Create user db2inst1 by typing the following:

   ```
   # mkuser pgrp=db2iadm1 db2inst1
   ```

4. Set default password for db2inst1 user as follows:

   When creating a user, the password is set to * (for invalid). It is necessary to set the password. To change the db2inst1 password, type the following:

   ```
   # passwd db2inst1
   Changing password for "db2inst1"
   db2inst1's New password: <your_password>
   Enter the new password again: <your_password>
   ```

5. Change the db2inst1 initial logon password.

   During the initial user logon, the user will be prompted to change a password. To change the initial logon password type the following:

   ```
   # login db2inst1
   db2inst1's Password: <your_password>
   db2inst1's New password: <your_password>
   Enter the new password again: <your_password>
   ```

6. Change ownership of /home/db2inst1 file system to user db2inst1 by typing the following:

   ```
   # chown -fR db2inst1:db2iadm1 /home/db2inst1
   ```

7. Verify file system ownership (db2inst1:db2iadm1)

   ```
   # ls -la /home/db2inst1
   ```

At this stage, we have created a logical volume for our database and created a username on the DB2 server machine that will be the DB2 instance name (for example, `db2inst1`). The DB2 server is now ready for the installation of DB2 V7.1 EE.

7.3.2 Install the DB2 V7.1 Server

To install the DB2 V7.1 EE Server for AIX, complete the following steps:

1. Log in to AIX as root and start a terminal session.
2. Mount the DB2 V7.1 CD-ROM as follows:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

3. Start DB2 install as follows:

   ```
   # cd /mnt
   # ./db2setup
   ```

4. When the Install DB2 V7.1 window appears, select **DB2 UDB Enterprise Edition**, highlight **OK** and press Enter.
5. When the Create DB2 Services window appears, select **Create a DB2 Instance**, highlight **OK** and press Enter.
6. DB2 instance authentication window appears, enter the following:
   - User Name: `db2inst1`
   - Group Name: `db2iadm1`
   - Home Directory: `/home/db2inst1`
   - Highlight **OK** and press Enter.

Note: The password field is intentionally left blank. In this example, we have manually created the user db2inst1 and set the password. DB2 can not change an existing user's password. If a password is entered it will be ignored.

7. The message `Cannot change password for an existing user - password will be ignored` is displayed. Highlight **OK** and press Enter.

8. When the Fence user window appears, enter the following:
   - User Name: `db2fenc1`
   - Group Name: `db2fadm1`
   - Home Directory: `/home/db2fenc1`
   - Password: `<your_password>`
   - Verify password: `<your_password>`
   - Highlight **OK,** and press Enter.
9. When the DB2 Warehouse Control Database window appears, select **Do not set up DB2 Warehouse Control Database**, highlight **OK** and press Enter.
10. When the DB2 Distributed Join for DB2 Data Sources window appears, select **Do not set up DB2 Distributed Join for DB2 Data Sources**, highlight **OK** and press Enter.
11. When the Create DB2 Services window appears, highlight **OK** and press Enter.
12. A Warning window displays the message `The Administration server is not created`. Highlight **OK** and press Enter.
13. The Summary Report window appears, listing the product components to be installed. Highlight **Continue** and press Enter.
14. The message `This is your last chance to stop` is displayed. Highlight **OK** and press Enter.

    The db2setup program installs the selected components. Depending on the speed of your processor, this can take up to 15 minutes. When the install completes, a notice window informs you whether it was successful.
15. A notice window displays `Completed successfully`. Highlight **OK** and press Enter.
16. Scan the Status Report to ensure that all components were installed successfully. Highlight **OK** and press Enter.
17. The DB2 Installer window appears. Highlight **Close** and press Enter.
18. A Warning message `The Administrative Server is not created` is displayed. Highlight **OK** and press Enter.
19. The message `Do you want to exit the DB2 Installer?` appears. Highlight **OK** and press Enter.
20. Unmount the CD-ROM as follows:

```
# cd /
# umount /mnt
```

21. Remove the DB2 V7.1 EE CD.

7.3.3 Install the DB2 V7.1 Fixpack 2a

To download and install DB2 UDB V7.1, Fixpack 2, complete the following steps:

1. Download the DB2 UDB V7.1 Fixpack 2a from the following:

   ```
   ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2aixv7/FP2a_U
   474808/
   ```

2. Install DB2 UDB V7.1 Fixpack 2a.

   For detailed information, refer to the Fixpack 2a readme.txt.

7.3.4 Create the WAS database on the DB2 Server

In preparation for the WebSphere Application Server installation, we will create a DB2 database called WAS (WebSphere Application Server repository database). The WAS database will be populated with a WAS schema and default values in a later step.

To create the WAS database manually, complete the following steps:

1. Log in as the DB2 instance owner as follows:

   ```
   # su - db2inst1
   ```

2. Type the following command to create the database:

   ```
   $ db2 create db was
   ```

   Note: The WebSphere Application Server repository database is often called WAS. However, the database can be named any valid DB2 database name.

3. Increase the application heap size for the WAS database as follows:

   ```
   $ db2 update db cfg for was using applheapsz 512
   $ exit
   ```

7.4 Install the WebSphere Application Server

This section describes how to install and configure the IBM WebSphere Application Server V3.5, Advanced Edition, WAS V3.5 Fixpack 2, and WAS V3.5.2 E-fixes required for WebSphere Commerce Suite V5.1.

The WebSphere Application Server installation is organized into the following phases:

- Install WAS V3.5
- Install WAS V3.5 Fixpack 2
- Install WAS V3.5.2 E-fixes

- Configure the WebSphere Application Server
- Verify the WebSphere Application Server

7.4.1 Install WAS V3.5

To install the IBM WebSphere Application Server V3.5, Advanced Edition, complete the following steps:

1. Log in to AIX as root and start a terminal session.
2. Insert the WebSphere Application Server CD into the CD-ROM drive (contains the HTTP Server).
3. Mount the CD-ROM:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

4. Change the working directory to the installation path:

   ```
   # cd /mnt/usr/sys/inst.images
   ```

5. Start the WAS V3.5 install by typing:

   ```
   # ./WebSphereInstallAIX.sh
   ```

   Note: WAS V3.5 provides two installation programs launched by the following scripts:

   - WebSphereInstallAIX.sh

     We used this interactive text mode install.

   - install.sh

     This GUI-based install requires a workaround to address a prereq check problem with DB2 V7.1 as follows:

     a. Copy WAS V3.5 to a directory on the local file system.

     b. Edit the prereq.properties file on the local file system. Change prereq_checker=1 to prereq_checker=0.

        This allows you to work around the problem. However, prereq checks are disabled (use with caution).

6. When prompted for the Installation Type, enter `2` for Custom Installation, and then press Enter.
7. When prompted for the directory where the WebSphere install packages are located, accept the default (/mnt/usr/sys/inst.images) and press Enter.
8. When prompted for security information, enter the following and press Enter:
   - User Name: `wasadmin`

- Password: <your_password>

9. When prompted for the WebSphere install packages, enter the number and then press Enter to select the package. Repeat this process until all of the following packages are selected:

   1. Application and Administrative Server
   2. Administrative Console
   3. Samples
   4. Help
   6. Configure Default Server and Applications
   7. WebServer Plugins

   When all the desired packages have been selected, press Enter to continue.

10. When prompted to enter the Web server plug-in to install type `1` for IBM HTTP Server V1.3.12.0 and then press Enter.
11. When prompted for language documentation, enter `1` for English and then press Enter.
12. When prompted for the database type, enter `1` for DB2 and then press Enter.
13. When prompted for the database name, enter `was` and press Enter.
14. When prompted to enter the database home director, enter `/home/db2inst1` and then press Enter.
15. When prompted to enter the database user, enter the following and then press Enter:
    - User: `db2inst1`
    - Password: <your_password>
    - Confirm Password: <your_password>
16. When prompted if you want to save the information to a file to perform a silent installation, enter `n` (no) and press Enter.
17. When the message `WebSphere will now begin the installation, press Enter to continue`, press Enter.
18. Tail the log file to check installation.

    ```
    # tail -f /tmp/WebSphere.instl
    ```

19. Unmount the CD-ROM.

The IBM WebSphere Application Server V3.5, Advanced Edition installation is now complete.

7.4.2 Install WAS V3.5 Fixpack 2

The WebSphere Application Server V3.5 Fixpack2 installation required by WCS V5.1. Fixpack 2a is included on the WCS V5.1 CD.

To install WAS V3.5 Fixpack 2a, complete the following steps:

1. Ensure the unzip utility is installed and in the path. Unzip is needed by the WAS V3.5 Fixpack 2 install.

   Note: The unzip utility can be found at:

   ```
   ftp://aixpdslib.seas.ucla.edu/pub/unzip/RISC/4.3/.
   ```

2. Ensure you are logged in as user ID root.
3. Stop the IBM HTTP Server:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl stop
   # ./adminctl stop
   ```

4. Insert the WebSphere Commerce Suite V5.1, Pro Edition CD.
5. Mount the CD-ROM:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

6. Create a directory with appropriate storage space for WAS V3.5 Fixpack 2 on your system.
7. Copy Fixpack 2 directory from the CD to your system:

   ```
   # cd /mnt/wasfix/wasfp2
   # cp -r * /<wasfp2_directory>
   ```

8. Change to the `<wasfp2_directory>` where you have copied Fixpack 2 by typing:

   ```
   # cd /<wasfp2_directory>
   ```

9. Start the WAS V3.5 Fixpack 2 install as follows:

   ```
   # ./install.sh
   ```

10. When prompted to enter the WebSphere root directory, enter `/usr/IBMWebAS` and then press Enter.
11. When prompted to enter whether you want to install the IHS Web Server PTF (y/n), enter `y` and press Enter.
12. When prompted to enter your webserver's doc root path, enter `/usr/HTTPServer/htdocs/en_US` (where en_US is your locale) and then press Enter.
13. When prompted with the message `Is this correct (y/n): /usr/HTTPServer/htdocs/en_US,` enter `y` if correct and then press Enter.

7.4.3 Install WAS V3.5.2 E-fixes

WebSphere Commerce Suite V5.1 requires that WebSphere Application Server V3.5 E-fixes be installed.

You can install the WebSphere Application Server V3.5.2 E-fixes manually or using the script that is available on the IBM Web site that will automate the WebSphere Application Server E-fixes installation. We recommend that you use the script since the manual installation method is tedious and will fail if a single error is made.

To use this script rather than the manual installation method, download the wcswasefix_unix.tar file from the following Web site:

```
http://www.ibm.com/software/webservers/commerce/wcs_pro/downloadsnc.html
```

To install the WAS V3.5.2 E-fixes, complete the following steps:

1. Insert the WebSphere Commerce Suite V5.1, Pro Edition for AIX CD into the CD-ROM drive on the WCS Server system. This CD includes the WAS V3.5.2 E-Fixes JAR files.
2. Create a download directory as follows:

   ```
   # mkdir -p <path_wasefix>
   ```

   Where <path_wasefix> is a directory that the files can be downloaded to.
3. Download the script from the following URL:

   ```
   http://www.ibm.com/software/webservers/commerce/wcs_pro/downloadsnc.html
   ```
4. Uncompress the tar file:

   ```
   # cd /<path_wasefix>
   # tar -xvf wcswasefix_unix.tar
   ```
5. Change permissions for script to make it executable:

   ```
   # chmod 777 wcswasefix_unix.sh
   ```
6. Run WAS E-fixes install

   ```
   # ./wcswasefix_unix.sh
   ```
7. Enter the source point of the E-fixes, `/mnt` (CD-ROM mount point), and then press Enter.

   Note: This will copy the E-fixes JAR files to the /usr/IBMWebAS/lib. In addition, the /usr/IBMWebAS/bin/admin.config file will be updated to include classpath entries for each JAR file.

8. Verify the following JAR files have been copied to the /usr/IBMWebAS/lib directory:
   - pq42952.jar
   - pq43040.jar
   - 888452_352.jar
   - 85699_352.jar
   - 88424.jar
   - sas-r3502-1115.jar
   - 88299.jar
9. Verify the admin.config classpath has been updated to include the JAR files listed in the previous step.

   Note: To install the E-fixes manually, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.

10. Copy xmlconfig.dtd:

```
# cd /usr/IBMWebAS/bin
# cp xmlconfig.dtd xmlconfig.35
# cp /mnt/wasfix/wasefix/WAS_3.5.2_efixes/xmlconfig.dtd /usr/IBMWebAS/bin
```

11. Modify the java.security file:
    a. Change to the /usr/IBMWebAS/java/jre/lib/security directory.
    b. Back up java.security to java.security.bak.
    c. Search for the following line in a text editor (for example, Wordpad):

```
security.provider.1=sun.security.provider.Sun
```

    d. Add this line below it:

```
security.provider.2=com.ibm.crypto.provider.IBMJCE
```

    e. Save the file.

The WebSphere Application Server V3.5.2, E-fixes installation is now complete.

7.4.4 Configure the WebSphere Application Server

This section provides instructions for configuring the WebSphere Application Server.

1. Ensure that the DB2 services are started.
2. Ensure that the IBM HTTP Server service is started.
3. Start the WebSphere Application Server.

```
# cd /usr/IBMWebAS/bin
# ./startupServer.sh
```

Note: The first time you start the WebSphere Application Server the default schema and configuration settings are populated. The parameter `initial.install.value=true` in the admin.config file tells the application server to initialize the database. Once the server is started, the value is set to false (`initial.install.value=false`).

You can recreate a clean WAS database by dropping the database, creating it, setting `initial.install.value=true`, and starting the IBM WS AdminServer service.

4. In a separate terminal window, tail the tracefile:

```
# cd /usr/IBMWebAS/logs
# tail -f tracefile
```

5. Review the messages in the `/usr/IBMWebAS/logs/tracefile` for errors. At the end of the tracefile, you should see the following message, which means the server has started successfully:

```
AdminServer   A WebSphere Administration server open for e-business
```

6. Start the WebSphere Application Server Administrative Console, in a new terminal window as follows:

```
# cd /usr/IBMWebAS/bin
# ./adminclient.sh
```

7. From the Administrator's Console, select **default_host**, and click the **Advanced** tab.

8. In the aliases field, ensure the aliases exist as seen in Table 7-6. If the aliases do not exist, add them by typing the alias in the Host Alias field, and then clicking **Apply**.

Table 7-6 WAS alias samples

Alias syntax	Alias example
<host.domain.com>	jganci2.itso.ral.ibm.com
<hostname>	jganci2
<ip_address>	9.24.104.217
<hostname.domain.com>:443	jganci2.itso.ral.ibm.com:443
<host>:443	jganci2:443
<ip_address>:443	9.24.104.217:443

7.4.5 Verify the WebSphere Application Server

This section provides instructions for verifying that the WebSphere Application Server has been successfully installed and configured.

1. Ensure the DB2 services are started.
2. Ensure the IBM HTTP Server service is started.
3. Ensure the IBM WebSphere Admin Server (startupServer.sh) is started.
4. Start the Default Server (application server) by doing the following:
   a. Start the WebSphere Application Server Administrator's Console:

      ```
      # cd /usr/IBMWebAS/bin
      # ./adminclient.sh &
      ```

   b. Select and expand **WebSphere Administrative Domain**.
   c. Select and expand your `<hostname>`.
   d. Select **Default Server**. Right-click **Start**.
   e. When the message `Default Server start completed successfully` is displayed, click **OK**.
5. From a Web browser enter the following URLs:

   ```
   http://<hostname>/servlet/snoop
   https://<hostname>/servlet/snoop
   ```

   If the snoop servlet executed correctly, the WebSphere Application Server is working properly.
6. Once the WebSphere Application Server has been verified, stop the Default Server (application server) from the WebSphere Administrative Console to conserve memory.

The WebSphere Application Server installation, configuration, and verification is now complete.

7.5 Install the WebSphere Commerce Suite

The WebSphere Commerce Suite installation will detect that DB2, HTTP Server, and the WebSphere Application Server are already installed, and then proceed.

Note: Prior to the installation of WebSphere Commerce Suite V5.1, Pro Edition for AIX, ensure that the prerequisites described in *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX* are met.

To install WebSphere Commerce Suite V5.1, Pro Edition for AIX, complete the following steps:

1. Log on to AIX as user ID root and start a terminal window.
2. Insert the WebSphere Commerce Suite V5.1, Pro Edition for AIX CD into the CD-ROM drive.
3. Mount the CD-ROM:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

4. Change to the WebSphere installation directory:

   ```
   # cd /mnt/WebSphereCommerceSuite
   ```

5. Start the installation as follows:

   ```
   # smitty install_all
   ```

6. In the input device/directory for software field, type `./` and press Enter.
7. Press F4 in the SOFTWARE to install field to list the installation components.
8. A list of the installation components is now presented. Make the selection of the appropriate software by using the down arrow and using the F7 key to make a selection. In our example, we selected the following components:
   - CommerceSuite.Base (all sub components)
   - CommerceSuite.Blaze
   - CommerceSuite.Docs

   Once all the selections have been made, press Enter.
9. To start the installation, press Enter.
10. When the confirmation message is displayed, press Enter.
11. When the installation is complete, scroll to the bottom of the display to confirm the installation was successful.
12. Unmount the CD-ROM:

    ```
    # cd /
    # umount /mnt
    ```

The WebSphere Commerce Suite V5.1, Pro Edition for AIX installation is now complete.

7.6 Create a WCS instance

After the installation of WebSphere Commerce Suite V5.1, Pro Edition for AIX and the prerequisite components, a WCS instance needs to be created. The WCS instance provides an environment for deploying and hosting a store. This section includes examples for creating a WCS instance using the Configuration Manager GUI or by using the command line utilities.

The section is organized as follows:

- Creating a WCS instance using the Configuration Manager
- Creating a WCS instance via the command line
- Verifying a WCS instance

7.6.1 Creating a WCS instance using the Configuration Manager

To create a WCS instance using the WCS Configuration Manager, complete the following steps:

1. Ensure the following servers are started:
   - IBM HTTP Server:

     ```
     # /usr/HTTPServer/bin/apachectl start
     ```

   - IBM DB2 Server:

     ```
     # su - db2inst1
     $ db2start
     ```

   - IBM WebSphere Admin Server:

     ```
     # /usr/IBMWebAS/bin/startupServer.sh
     ```

2. Ensure that the config_env.sh includes the correct DB2 instance name as follows:
   a. Change to the directory of the config_env.sh script file:

      ```
      # cd /usr/lpp/CommerceSuite/bin
      ```

   b. Ensure that the DB2INSTANCE=<your_db2_instance>:

      ```
      export DB2INSTANCE=db2inst1
      ```

      Where `db2inst1` is the name of <your_db2_instance>.

3. Start the Configuration Manager Server as follows:

   ```
   # cd /usr/lpp/CommerceSuite/bin
   ```

   # ./config_server.sh

   Note: Do not execute this command as a background process (&) to avoid a security risk.

4. Start the Configuration Manager (client) as follows:

```
# cd /usr/lpp/CommerceSuite/bin
# ./config_client.sh
```

5. When the Config Authentication window appears, enter the following and then click **OK**.:
   - User ID: `webadmin` (default)
   - Password: `webibm` (default)

   Note: The first time you log in, you will be prompted with the message `For security purposes, please change your default CM password.`

   The Configuration Manager should now be displayed.

   In the remaining steps, we provide examples for our runtime environment settings. Refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX* for all the possible configuration options and values.
6. Select and expand your `<hostname>` from the Configuration Manager window.
7. Select **Instance List**. Right-click **Instances** -> **Create Instance**.
8. When the Select an Option window appears with the message `Do you want to use an existing instance as the basis for this instance?`, click **No**.

   Note: The Instance Creation Wizard window should appear. This wizard will guide you sequentially through each tab to configure components. You can go back to any of these tabs directly by clicking on it.

9. Select the **Instance** tab (default starting point), and enter the following and then click **Next**:
   - Instance name: `wcs`
   - Instance root path: `/usr/lpp/CommerceSuite/instances/wcs`

     Note: You must manually change the instance name. By default the directory will be demo.

   - Deselect the **Merchant Key** checkbox.
   - Merchant Key: `<your_merchant_key>`

     The merchant key requires a 16-character hexadecimal value. This is a user-defined field.

Note: To avoid a security risk, you must enter your own defined value for the merchant key.

10. On the Database tab, enter the following and then click **Next**:
    - Database administrator name: `db2inst1`
    - Database administrator password: `<db2inst1_password>`
    - Database name: `wcs`
    - Database type: select `DB2` from the pull-down
    - Database user name: `db2inst1`
    - Database user password: `<db2inst1_password>`
    - Check **Set as active database** (default)

    Note: The Use remote database option must be checked if your are using a remote database (for example, a 3-tier configuration). Refer to 7.8, “Installing a 3-tier runtime environment” on page 179 for details.

11. On the Languages tab, accept the default (wcs.bootstrap_multi_en.xml), and click **Next**.
12. On the Web Server tab, enter the following and then click **Next**:
    - Hostname: `<hostname>`

      It should detect this automatically. If not, enter the fully qualified hostname, for example `jganci2.itso.ral.ibm.com`.
    - Web Server Type: select **IBM HTTP Server** from pull-down
    - Primary Document Root: `/usr/HTTPServer/htdocs/en_US`
    - Server Port: `80`
13. On the WebSphere tab, do the following and then click **Next**:
    - Data Source Name: `WebSphere Commerce Suite DB2 DataSource wcs`

      The wcs instance name is added at the end of this string to create the data source name.
    - Port Number: `900` (default)
    - JDBC Driver Location: `/usr/lpp/db2_07_01/java12/db2java.zip`
    - Check **Stores Web Application** (default)
    - Check **Tools Web Application** (default)
14. On the Payment Manager tab, accept the defaults, and click **Next**.

In this example, we do not configure Payment Manager.

15. On the Log System tab, accept the defaults, and click **Next**.
16. On the Messaging tab, accept the defaults, and click **Finish**.
17. When the Population Confirmation window appears, click **Yes**.

    Note: The instance creation process takes approximately 10-20 minutes depending on your system specifications.

18. When the Configuration Manager is done creating the instance, you should see a message `Instance was successfully created`. Click **OK** and close the WCS Configuration Manager.

    Note: We recommend that you stop the Configuration Manager Server after the WCS instance is created to avoid a security risk and reduce memory consumption.

19. Restart the WebSphere Commerce Server.

    After the WCS instance is created, the WebSphere Commerce Server - <wcs_instance> (application server), must be restarted by completing the following steps:

    a. Start the WebSphere Administrator's Console as follows:

    ```
    # cd /usr/IBMWebAS/bin
    # ./adminclient.sh
    ```

    b. Select and expand **WebSphere Administrative Domain** -> **<your_host>**.

    c. Select **WebSphere Commerce Server - wcs**. Right-click **Stop**. Once the server has stopped, right-click **Start**.

The Configuration Manager WCS instance creation is now complete. Proceed to 7.6.3, "Verifying a WCS instance" on page 172.

7.6.2 Creating a WCS instance via the command line

This section describes how to create a WCS instance using the command line utilities as an alternative to creating a WCS instance using the Configuration Manager. This method of instance creation is necessary for environments that do not have a system console attached to the systems. In addition, WCS instance creation via the command line offers the ability to automate tasks.

Note: This section should be read in conjunction with Chapter 20 in the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*, which includes information on command-line instance creation.

The procedure to create a WCS instance via the command line is organized as follows:

- Create a directory for your WCS instance
- Create a WCS instance XML configuration file
- Modify the WCS instance XML configuration file
- Create the WCS instance using the config_client.sh script

Create a directory for your WCS instance

Each WCS instance uses requires its own directory structure to store its configuration files. When using the Configuration Manager, this structure is kept in the /usr/lpp/CommerceSuite/instances/<instance_name> directory and managed for you. When creating a WCS instance via the command line, the directory must be created manually.

To create a directory for the WCS instance, complete the following steps:

1. Log in to AIX as root and start a terminal session.
2. Create a directory for the instance as follows:

   ```
   # cd /usr/lpp/CommerceSuite/instances
   # mkdir wcs
   ```

   Where `wcs` is the name of the WCS instance directory.

Create a WCS instance XML configuration file

There are two choices for creating a WCS instance XML configuration file:

- Create your own version based on an existing WCS instance.
- Copy the /usr/lpp/CommerceSuite/instances/default/xml/default.xml file.

Note: In this example, we modified an existing WCS instance XML configuration file created by using the Configuration Manager.

Copy the WCS instance XML configuration file to the root <wcs_instance>\xml directory as follows:

1. Log in to AIX as root and start a terminal session.

2. Create an XML directory in the WCS instance directory:

```
# cd /usr/lpp/CommerceSuite/instances/wcs
# mkdir xml
```

3. Copy the WCS instance XML configuration file to the WCS instance XML directory:

```
# cp wcs.xml /usr/lpp/CommerceSuite/instances/wcs/xml/wcs.xml
```

   Ensure that the WCS instance XML configuration file has the same name as the instance.

Modify the WCS instance XML configuration file

If you copied an existing instance file, a number of changes are required to be made prior to completing the instance creation. In our example, we intentionally used the Configuration Manager to create our WCS instance XML configuration file with the exact settings we needed for our environment.

The WCS instance XML configuration file contains the encrypted password for the DB2 instance owner.

To update the WCS instance XML configuration file with the correct encrypted DB2 instance owner password, complete the following steps:

1. Change to the WCS working directory and run the script file as follows:

```
# cd /usr/lpp/CommerceSuite/bin
# ./wcs_encrypt.sh db2inst1
```

   Where `db2inst1` is the DB2 instance owner password you wish to encrypt.

2. The script file publishes the value of the password in an encrypted form. Insert the encrypted password string into the WCS instance XML configuration file.

   Note: Ensure the complete string is copied. An incorrect password will result in the WCS instance not being created correctly.

Create the WCS instance using the config_client.sh script

To create the WCS instance using the config_client.sh script file, complete the following steps:

1. Log in to AIX as root and start a terminal session..
2. In the terminal window, start the configuration server:

```
# cd /usr/lpp/CommerceSuite/bin
# ./config_server.sh
```

3. Once started, in another window, start the command line instance creation:

```
# cd /usr/lpp/CommerceSuite/bin
# ./config_client.sh - startCmdLineConfig addInstance
/usr/instances/prod/prod.xml
```

Note: This will start the instance creation process, which take several minutes to complete. Once complete, the instance is ready to use.

There are other command line parameters that can be passed to `config_client.sh`. These options are documented in Chapter 20 of the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.

7.6.3 Verifying a WCS instance

To verify the WCS instance is created properly, complete the following steps:

- Verify the existence of the instance xml configuration file <instance_name>.xml in the following directory:

  `/usr/lpp/CommerceSuite/instances/<instance_name>/xml/<instance_name>.xml`

  For example:

  `/usr/lpp/CommerceSuite/instances/wcs/xml/wcs.xml`

- Review the contents of the instance creation log createdb.log in the following directory:

  `/usr/lpp/CommerceSuite/instances/<instance_name>/logs`

  For example:

  `/usr/lpp/CommerceSuite/instances/wcs/logs/createdb.log`

- Review the contents of the wcs.log regarding the deployment of WCS within the WebSphere Application Server database repository, found in the following directory:

  `/usr/lpp/CommerceSuite/instances/<instance_name>/logs`

  For example:

  `/usr/lpp/CommerceSuite/instances/wcs/logs/wcs.log`

- Verify the WebSphere Commerce Suite Administration Console starts by entering the following URL in a Web browser:

  `http://<hostname>/adminconsole`

Note: The first time this tool is loaded, it takes approximately 1-2 minutes to compile the JSPs. By default, the user ID and password are wcsadmin.

7.7 Deploy the sample store

To further verify the WCS runtime, we recommend that you deploy the sample store called InFashion, supplied with WCS V5.1, using Store Services.

Note: This step is required if you intend to use this environment for development. The WebSphere Commerce Studio requires that the InFashion store be published and be named InFashion.

7.7.1 Publish store from Store Services

To deploy the sample store using Store Services, complete the following steps:

1. Start Internet Explorer V5.5 on a Windows system (browser type and level required by WCS).
2. To start Store Services, enter the following URL on an Internet Explorer V5.5 Web browser: `http://<hostname>/storeservices.`
3. A Security Alert window may appear. Accept the certificate and click **Yes** to continue.
4. When the Store Services Logon window appears, enter the following:
   - User name: `wcsadmin`
   - Password: `wcsadmin` (default)
   - Click **Log On**

   Note: The first time you log on, it takes a while to compile the JSPs.
5. When the Create Store Archive window appears, enter the following:
   - In the sample pull-down select **infashion_en_US_es_ES.sar**.

     This sample includes multicultural support for US English and Spanish for Spain locales. You can select any of the sample InFashion stores for your locale needs.
   - Store archive (required): `InFashion`

     This is the name of the SAR file that will be generated to publish to your runtime environment. For example, InFashion.sar will be created.
   - Store directory (required): `InFashion`

     This is the directory that will serve as the root for this store.
   - Store owner: Default Organization (default)

   Click **OK**.

   You should see a message, `InFashion.sar created successfully`. Click **OK**.

6. When the Store Archive window appears, take note of the Publish Status column of the table. The status will be Not published.
   a. Check the **InFashion.sar** checkbox under the heading Store archive.
   b. Click **Publish**.
7. When the Publish Store Archive window appears, accept the defaults and click **OK**.

   Note: Publishing a store takes approximately 10-15 minutes.

8. There are several publishing status states:
   – Not published
   – Awaiting publishing
   – Publishing
   – Published completed successfully

   The first state will be Awaiting publishing. Click **Refresh**. The publishing status will change to Publishing. If successful, the status will change to Publish completed successfully.

   After approximately 10 minutes, you may want to click **Refresh** to see if the publishing status has changed.
9. When the status changes to Publish completed successfully, check the checkbox next to **InFashion.sar**, and then click **Publish Summary**.
10. When the Publish Summary window appears, click **Launch Store** to launch the sample store.
11. You will be prompted to enter the Web application Web path. Accept the default (/webapp/wcs/stores) and click **OK**.

    Note: The first time the store loads it will take a while due to the compilation of store JSPs.

12. Take note of the sample store URL (bookmark or add to favorites):

    ```
    https://<hostname>/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&langId=-1&catalogId=10001
    ```

    Note: The sample store URL launched from Store Services is an HTTPS (secure) home page. You may consider creating a bookmark to the HTTP (non-secure) Web page.

13. Close Store Services.

7.7.2 Deploy the sample store via the command line

As an alternative to deploying the store using Store Services, the store can be deployed via the command line using a WCS script file. This section details the steps to deploy a store using the command line.

To publish using the command line, complete the following steps:

1. On the WCS/WAS server, log in to AIX as root and create a terminal window.
2. In this terminal window, issue the following commands to obtain the current list of store and catalog IDs being used:

   ```
   # su - db2inst1
   $ db2 connect to <store_database_name> user db2inst1 using <password>
   $ db2 select * from store
   $ db2 select * from catalog
   ```

   Document the values used for the store and catalog.
3. Copy the store SAR file to the following directory:

   ```
   # cp <store_archive.sar> /usr/lpp/CommerceSuite/stores/web
   ```

   Where `<store_archive.sar>` is the SAR file for the store to publish.
4. The AIX script file publishstore.sh in the /usr/lpp/CommerceSuite/bin is used to publish the store. It requires eight parameters, listed in Table 7-7.

Table 7-7 Publishstore.sh parameter description

Parameter Name	Description / Value
SAR	The full pathname of the SAR file. For example: `/us/lpp/CommerceSuite/stores/web/mystore.sar`
svr	Hostname of the WCS server.
USERID	Logon ID for the WebSphere Commerce Suite, for example, `wcsadmin`.
PWD	Password for the above user ID.
MODE	`insert` or `update`
CONFIG	The XML configuration file of the WCS instance. For example: `/usr/lpp/CommerceSuite/instances/<instance_name>/xml<instance_name.xml>`

Parameter Name	Description / Value
XML	The list of XML files in the SAR to be published. To publish all, use `ALL`. To publish everything except for the catalog, use `NOCATLG`.
ASSET	The list of asset files in the SAR. For example, webapp.zip and the paths to which they will be published. For multiple asset files, list them all, separated by a space. For example: `/usr/lpp/CommerceSuite/stores/web=webapp.zip`

Note: In a 3-tier environment, the value of `svr`, will be the HTTP Server machine name, *not* the WCS/WAS server.

5. When editing the file, place a comment character # in the script file as shown in Example 7-1.

Example 7-1 setevn.sh script

```
#!/bin/ksh
. $PWD/setenv.sh

#if [[ "$8" = "" ]] ; then
#
#  echo 'Usage:'
#  echo '  $1 sar file name with full path'
#  echo '  $2 hostname of server running wcs'
#  echo '  $3 wcs username'
#  echo '  $4 wcs password'
#  echo '  $5 mode of process'
#  echo '  $6 config file name with full path'
#  echo '  $7 list of xml files in SAR to be processed'
#  echo '  $8 list of asset files in SAR and the paths that they are assigned
to \n '
#  exit 1
#fi
```

This will allow the script to run without checking the syntax of the command line.

6. Once these changes have been made, run the shell script:

```
# ./publishstore.sh
```

This should report messages that the store is being published, and that publishing has been successful.

Once the store has been published, repeat the commands from step 2 to see the new store-id and catalog-id values.

7. Issue the following commands to obtain the added list of store and catalog IDs created:

```
# su - db2inst1
$ db2 connect to <store_database_name> user db2inst1 using <password>
$ db2 select * from store
$ db2 select * from catalog
```

Document the values used for the store and catalog.

Note: At the time of writing this redbook, this script reported a number of errors, which were not able to be resolved. As a result of this, it is recommended that the store be published using Store Services.

7.7.3 Creating an alias for the store

To make it easier for users, including yourself, to access your store, we recommend that you create an alias and an index.html file containing the location.href to your store.

Allow quick access to your store from a Web browser by typing the following URL:

- `http://<hostname>/<alias>`

  For example: `http://jganci2/mystore`

Instead of:

- `http://<hostname>/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&langId=-1&catalogId=10001`

Create an index.html with a href for your store

To create an index.html file with the location.href for your store, complete the following steps:

1. Change to the directory of your published store. For example:

```
# cd /usr/lpp/CommerceSuite/stores/web
```

2. Create an index.html file in the root of the directory created in the previous step. Refer to Example 7-2 for a sample index.html.
3. Change the location.href to match your store settings. For example, change the hostname.

4. Set permissions on the index.html file (for example, `chmod 755 index.html`).

Example 7-2 Sample index.html

```
<html>
<head>
<title>InFashion sample store</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
location.href =
"http://jganci2.itso.ral.ibm.com/webapp/wcs/stores/servlet/StoreCatalogDisplay?
storeId=10001&langId=-1&catalogId=10001";
</script>
</head>
<body bgcolor="#FFFFFF">
<b><font size="1" face="Arial, Helvetica, sans-serif" color="#333333">InFashion
sample store
<br>One moment.....</font></b>
</body>
</html>
```

Add the alias to the httpd.conf

To add an alias to the httpd.conf, complete the following steps:

1. Stop the IBM HTTP Server:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl stop
   ```

2. Change to the directory of the httpd.conf file:

   ```
   # cd /usr/HTTPServer/conf
   ```

3. Modify the httpd.conf file. Add the following alias at the end of the httpd.conf file (path of your published store):

   ```
   Alias /mystore "/usr/lpp/CommerceSuite/stores/web/"
   ```

   Note: After the WCS V5.1 install, the httpd.conf file no longer is editable using Notepad (removal of crlf). We used Wordpad to edit the httpd.conf.

4. Start the IBM HTTP Server:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl start
   ```

Test the alias

Once you have updated the httpd.conf, created the index.html file, and stopped and restarted the IBM HTTP Server, you can now test the alias. Enter the following URL in a Web browser to access your store:

- http://<hostname>/<alias>

  For example: http://jganci2/mystore

7.8 Installing a 3-tier runtime environment

As described in 5.2.3, “High-level installation steps: 3-tier” on page 73, this 3-tier WCS runtime environment working example starts from a 3-tier WebSphere Application Server configuration. This approach to creating a 3-tier runtime may be better suited for WebSphere Application Server experts. By adding a third system, Web server with the WebSphere plug-in, separate from the WebSphere Application Server, we are able to configure OSE remote to build a 3-tier WCS runtime environment. Many of the component installations are the same as defined for the 1-tier configuration and will be referenced where possible.

Some advanced configuration techniques are included, such as OSE remote configuration, IPSec configuration, manual WCS database creation on the database server, and manual configuration of the remote Web Server.

In the new 3-tier configuration, the systems are defined as follows:

- Machine A: Web server

  For example, IBM HTTP Server and the WebSphere plug-in.

- Machine B: Commerce/WebSphere Application Server

  For example, WebSphere Application Server, WebSphere Commerce Suite, DB2 client.

- Machine C: Database server

  For example, DB2 server.

The 3-tier WCS runtime environment for AIX installation and configuration, is organized into the following phases:

- Step 1: Install AIX and prerequisites
- Step 2: Install the DB2 server
- Step 3: Install the DB2 client
- Step 4: Install the WebSphere Application Server
- Step 5: Install the HTTP Server
- Step 6: Install the WebSphere plug-in
- Step 7: Configure OSE on the WebSphere Application Server
- Step 8: Configure OSE on the HTTP Server

- Step 9: Install the WebSphere Commerce Suite
- Step 10: Copy the loader package
- Step 11: Create the WCS database manually
- Step 12: Create a WCS instance
- Step 13: Add the WCS entries to the httpd.conf
- Step 14: Deploy the sample store
- Step 15: Serve static content from the Web server
- Step 17: Configure the IPSec tunnel

7.8.1 Step 1: Install AIX and prerequisites

Install AIX and prerequisites on the Web Server (machine A), Commerce/WebSphere Application Server (machine B), and Database Server (machine C).

7.8.2 Step 2: Install the DB2 server

Install the DB2 UDB V7.1, EE for AIX on the Database Server (machine C).

Refer to 7.3, “Install the DB2 Server” on page 151 for detailed information.

7.8.3 Step 3: Install the DB2 client

Install the IBM DB2 UDB V7.1, EE for AIX, Administrative Client (DB2 client), on the Commerce/WebSphere Application Server (machine B).

The DB2 client installation is organized into the following phases:

- Install the DB2 client
- Configure the DB2 client and server connectivity
- Verify the DB2 client and server connectivity

Install the DB2 client

To install the IBM DB2 UDB V7.1, EE Administrative Client (DB2 client), complete the following steps on the Commerce/WebSphere Application Server:

1. Log in to AIX as root and start a terminal session.
2. Insert the IBM DB2 UDB V7.1, EE for AIX CD into the CD-ROM drive.
3. Mount the CD-ROM:

```
# mount -r -v cdrfs /dev/cd0 /mnt
```

4. Start DB2 install as follows:

```
# cd /mnt
# ./db2setup
```

5. When the Install DB2 V7.1 window appears, select **DB2 Administrative Client**, highlight **OK** and press Enter.
6. When the Create DB2 Services window appears, select **Create a DB2 Instance**, highlight **OK** and press Enter.
7. When the DB2 instance authentication window appears, enter the following sample values, highlight **OK** and press Enter.
   - User Name: `db2inst1`
   - Group Name: `db2iadm1`
   - Home Directory: `/home/db2inst1`
   - Password: `<your_password>`
   - Verify password: `<your_password>`
8. When the Fence user window appears, enter the following sample values, highlight **OK** and then press Enter:
   - User Name: `db2fenc1`
   - Group Name: `db2fadm1`
   - Home Directory: `/home/db2fenc1`
   - Password: `<your_password>`
   - Verify password: `<your_password>`
9. When the Create DB2 Services window appears, highlight **OK** and press Enter.
10. A Warning window displays the message `The Administration server is not created`. Highlight **OK** and press Enter.
11. The Summary Report window appears, listing the product components to be installed. Highlight **Continue** and press Enter.
12. The message `This is your last chance to stop` is displayed. Highlight **OK** and press Enter.

    The db2setup program installs the selected components. Depending on the speed of your processor, this can take up to 15 minutes. When the install completes, a notice window informs you whether it was successful.
13. A notice window displays `Completed successfully`. Highlight **OK** and press Enter.
14. Scan the Status Report to ensure that all components were installed successfully. Highlight **OK** and press Enter.

15. The DB2 Installer window appears. Highlight **Close** and press Enter.
16. A Warning message `The Administrative Server is not created` is displayed. Highlight OK and press Enter.
17. The message `Do you want to exit the DB2 Installer?` appears. Highlight **OK** and press Enter.

Install DB2 V7.1 Fixpack 2a

For details, refer to 7.3.3, “Install the DB2 V7.1 Fixpack 2a” on page 157.

Configure the DB2 client and server connectivity

To configure the DB2 client and server, complete the following steps on the Commerce/WebSphere Application Server (machine B):

1. Catalog the Database Server TCP/IP node as follows:

```
# su - db2inst1
$ db2 catalog tcpip node dbsrvaix remote dbsrvaix server 50000
```

Syntax:

```
> db2 catalog tcpip node <node_name> remote <db_server_hostname> server
50000
```

2. Catalog the WAS database as follows:

```
$ db2 catalog db was at node rs600013
```

Syntax:

```
> db2 catalog db <database_name> at node <node_name>
```

Verify the DB2 client and server connectivity

To verify the DB2 client and server connectivity, complete the following steps on the Commerce/WebSphere Application Server (machine B):

1. Verify the attach to the TCP/IP node as follows:

```
$ db2 attach to rs600013 user db2inst1 using <your_password>
```

Syntax:

```
> db2 attach to <node_name> user <db2inst_ID> using <db2inst_password>
```

2. Verify the connect to database from the Commerce/WebSphere Application Server (machine B) to the Database Server (machine C) as follows:

```
$ db2 connect to was user db2inst1 using <your_password>
$ exit
```

Syntax:

```
> db2 connect to <db_name> user <db2inst_ID> using <db2inst_password>
```

7.8.4 Step 4: Install the WebSphere Application Server

The IBM WebSphere Application Server V3.5, Advanced Edition for AIX, WAS V3.5 Fixpack 2, and WAS V3.5.2 E-fixes are installed on the Commerce/WebSphere Application Server (machine B).

Refer to 7.4, “Install the WebSphere Application Server” on page 157, for detailed information.

7.8.5 Step 5: Install the HTTP Server

The IBM HTTP Server V1.3.12 for AIX (Apache) is installed on the Web Server (machine A).

Refer to 7.2, “Install the IBM HTTP Server” on page 144, for detailed information.

7.8.6 Step 6: Install the WebSphere plug-in

The WebSphere Application Server plug-in for the IBM HTTP Server V1.3.12 for AIX is installed on the Web Server (machine A). In order for the Web Server (machine A) to communicate with the Commerce/WebSphere Application Server (machine B), the WebSphere plug-in must be installed. The plug-in is used to provided communication between the HTTP Server and the WebSphere Application Server using the Open Servlet Engine (OSE) protocol.

WebSphere Application Server plug-in install

To install the WebSphere Application Server plug-in and components required to support OSE remote, complete the following steps:

1. Ensure that AIX and prerequisites for WAS have been installed.
2. Ensure that the IBM HTTP Server is installed and configured with SSL support.
3. Log in to AIX as root and start a terminal session.
4. Stop the IBM HTTP Server as follows:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl stop
   ```

5. Insert the WebSphere Application Server CD into the CD-ROM drive (contains the HTTP Server).

6. Mount the CD-ROM:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

7. Change the working directory to the installation path:

   ```
   # cd /mnt/usr/sys/inst.images
   ```

8. Start the WAS V3.5 install by typing:

   ```
   # ./WebSphereInstallAIX.sh
   ```

9. When prompted for the Installation Type, enter `2` for Custom Installation, and then press Enter.
10. When prompted for the directory where the WebSphere install packages are located, accept the default (/mnt/usr/sys/inst.images) and press Enter.
11. When prompted for security information, enter the following and press Enter:
    - User Name: `wasadmin`
    - Password: `<your_password>`
12. When prompted for the WebSphere install packages, enter the number and then press Enter to select the package. Repeat this process until the following packages are selected:
    - 1. Application and Administrative Server
    - 7. WebServer Plugins

    When the desired packages have been selected, press Enter to continue.
13. When prompted to enter the Web server plug-in to install type `1` for IBM HTTP Server V1.3.12.0 and then press Enter.
14. When prompted for language documentation, enter `1` for English and then press Enter.
15. When prompted for the database type, enter `1` for DB2 and then press Enter.
16. When prompted for the database name, enter `was` and press Enter.
17. When prompted to enter the database home director, enter `/home/db2inst1` and then press Enter.
18. When prompted to enter the database user, enter the following and then press Enter:
    - User: `db2inst1`
    - Password: `<your_password>`
    - Confirm Password: `<your_password>`
19. When prompted if you want to save the information to a file to perform a silent installation, enter `n` (no) and press Enter.

20. When the message `WebSphere will now begin the installation, press Enter to continue`, press Enter.
21. Tail the log file to check installation.

```
# tail -f /tmp/WebSphere.instl
```

22. Unmount the CD-ROM.

WAS V3.5 Fixpack2

Refer to 7.4.2, “Install WAS V3.5 Fixpack 2” on page 160 for detailed information.

WAS V3.5.2 E-fixes

Refer to 7.4.3, “Install WAS V3.5.2 E-fixes” on page 161 for detailed information.

7.8.7 Step 7: Configure OSE on the WebSphere Application Server

To configure OSE remote on the Commerce/WebSphere Application Server (machine B), complete the following steps:

1. Ensure that WebSphere Admin Server is started:

```
# cd /usr/IBMWebAS/bin
# ./startupServer.sh
```

2. Start the WebSphere Application Server Administrator’s Console.

```
# cd /usr/IBMWebAS/bin
# ./adminclient.sh &
```

3. Configure the WebSphere Application Server servlet engine to use INET Sockets, by completing the following steps:
   a. From the tree view in the Administrator’s Console, expand **Hostname** -> **Default Server** -> **<your_host>** -> **Default Server** -> **Default Servlet Engine**.
   b. Click the **Advanced** tab, and then click **Settings**.
   c. In the Edit Servlet Engine Transport window, select **INET Sockets** from the Transport Type drop-down list, and then click **OK**.
   d. Click **Apply**.
   e. The changes will not take effect until you stop and start the application server.
4. Add the aliases of each Web server (machine A) to the alias list of the virtual host for the application server. In order to access WebSphere Application Server applications, such as WCS stores using SSL, you need to add virtual hosts.

Note: Every time you add virtual hosts you will need to rerun the OSE remote utility to regenerate the properties files for OSE remote configuration:

a. From the tree view, select **default_host**, and then click the **Advanced** tab.

b. Add the virtual host alias's host name and IP address for each Web server. For ports other than port 80 (the default), add the port (for example, if you are using SSL, add <host_name>:443, x.x.x.x:443).

c. Press Enter after you enter each entry, and click **Apply** when you are finished to save your changes.

5. Stop and start the WebSphere Application Server, by stopping and starting the WebSphere Admin Server (startupServer.sh) on machine B.

   Wait for the product to refresh the automatically generated plug-in configuration files. This could take up to a few minutes.

   Verify that a refresh has occurred by checking the time stamp in any of the three properties files in the temp directory under the WebSphere Application Server product installation directory. The files tell the WebSphere plug-in for the Web server how to find the remote application servers.

7.8.8 Step 8: Configure OSE on the HTTP Server

This section describes the procedure to configure the OSE remote plug-in on the HTTP Server. This plug-in accepts JSP and servlet requests and redirects them to the WebSphere Application Server. The HTTP Server is responsible for serving static HTML pages.

To configure OSE remote on the HTTP Server (machine A), complete the following steps:

1. Log in to AIX as root and start a terminal session.
2. Stop the IBM HTTP Server and the IBM HTTP Administration Server as follows:

```
# cd /usr/HTTPServer/bin/
# ./apachectl stop
# ./adminctl stop
```

3. Execute the OSERemoteconfig.sh script as follows:

```
# cd /usr/HTTPServer/bin
# ./OSERemoteConfig.sh -adminNodeName pluto
```

Syntax:

```
> ./OSERemoteConfig.sh -adminNodeName WAS-MACHINE-NAME
-nameServiceNodeName WAS-MACHINE-NAME
```

- Where *WAS-MACHINE-NAME* is your WebSphere Application Server hostname.
- `-nameServiceNodeName` is optional.

The OSERemoteConfig.sh script file updates the properties file in the /usr/IBMWebAS/temp directory based on the current settings in the WAS repository database. These files are then loaded when the WAS bootstrap.properties is read by the HTTP Server. The properties file names and description can be found in Table 7-8.

Table 7-8 WAS bootstrap properties

Filename	Description
queues.properties	Details which application servers are running on which machines, and which OSE ports they are using.
rules.properties	Details which URLs and virtual hosts aliases correspond to a given application server.
vhosts.properties	Describes which IP address and hostnames correspond to a given application server virtual host alias.

Important: If you make changes to the WebSphere Application Server configuration, you must re-run the OSRemoteConfig.sh script, so that the HTTP Server understands the new WebSphere Application Server configuration.
You will see a large number of information messages displayed when you run the OS-RemoteConfig script. These are informational only, and indicate the Websphere environment "appears consistent".

4. Once the OSRemoteConfig.sh script has completed, you will need to re-start the HTTP Server as follows:

```
# cd /usr/HTTPServer/bin
# ./apachectl start
```

5. Verify the OSE remote configuration.

   The HTTP Server is now configured for OSE remote with the WebSphere Application Server. By entering the following URL in a Web browser, we can verify the configuration:

```
http://<webserver_hostname>/webapp/examples/ShowConfig
```

   The serving of pages and servlets demonstrates that the components are working correctly. The output of this URL should show the servlet engine configuration for the WebSphere Application Server in your browser.

Note: If the output in the Web browser indicates an internal server error has occurred, this could mean the WebSphere Application Server has not started, or not completed its starting cycle.

7.8.9 Step 9: Install the WebSphere Commerce Suite

Install WebSphere Commerce Suite V5.1, Pro Edition for AIX on the Commerce/WebSphere Application Server.

Refer to 7.5, “Install the WebSphere Commerce Suite” on page 164 for detailed information.

7.8.10 Step 10: Copy the loader package

At the time of writing this redbook, we found the WCS instance creation failed due to a loader utility failure in remote database configurations. To work around this problem, your DBA requires that the WCS database be created manually on the Database Server. This is accomplished by zipping up the loader package files, copying the zip to the Database Server, setting up the loader package, and then creating the WCS database manually. There are a number of files and directories required to be duplicated on the Database Server (machine C) from the Commerce/WebSphere Application Server (machine B).

Note: As an alternative to copying the directories and their contents from the Commerce/WebSphere Application Server (machine B) to the Database Server (machine C), install the WebSphere Application Server and WebSphere Commerce Suite on the Database Server. Neither WAS or WCS e would need to be started on this machine.

Zip the loader package files

To zip the loader package and associated files, complete the following steps:

1. Log on to the Database Server (machine C) as the user ID root, and start a terminal window.
2. Create the following directory structure on the Database Server:

```
# mkdir -p /usr/lpp
# mkdir -p /usr/lpp/CommerceSuite
# mkdir -p /usr/IBMWebAS
# mkdir -p /usr/IBMWebAS/java
```

3. On the WAS machine, export the /usr/ directory, by using “smitty”.
4. On the Database machine, mount the WAS /usr directory:

```
# mount rs600012:/usr /mnt
```

5. On the Database machine, copy the contents of the /usr/lpp/CommerceSuite (instances, xml, bin, lib, and schema directories that are on the WAS machine (/mnt mount point). There is a large amount of data to copy, so ensure there is enough temporary disk space available.

```
# cd /mnt/lpp/CommerceSuite
# tar cvf /usr/lpp/CommerceSuite/instances.tar instances
# tar cvf /usr/lpp/CommerceSuite/xml.tar xml
# tar cvf /usr/lpp/CommerceSuite/lib.tar lib
# tar cvf /usr/lpp/CommerceSuite/bin.tar bin
```

```
# tar cvf /usr/lpp/CommerceSuite/schema.tar schema
# cd /mnt/IBMWebAS/java
# tar cvf /usr/IBMWebAS/was_java_lib.tar lib
# cd /mnt/IBMWebAS/java/jre
# tar cvf /usr/IBMWebAS/java/java_extra.tar lib
```

6. Un-tar the files as follows:

```
# cd /usr/lpp
# mkdir CommerceSuite
# tar xvf xml.tar
# tar xvf lib.tar
# tar xvf bin.tar
# tar xvf schema.tar
# cd /usr/IBMWebAS
# tar xvf was_java_lib.tar
# cd /usr/IBMWebAS/java
# tar xvf /usr/IBMWebAS/java/java_extra.tar
```

7. On the Database machine, alter the permissions on the new directory structures to be 777. This is a temporary measure. The command to complete this is:

```
# chmod -R 777 /usr/lpp/CommerceSuite
# chmod -R 777 /usr/IBMWebAS/
```

7.8.11 Step 11: Create the WCS database manually

Once the loader utility has been packaged and set up on the Database Server (machine C), the WCS database can be created manually in preparation for the WCS instance creation.

To create the WCS database manually on the Database Server (machine C), complete the following steps:

1. Log on as user ID root and open a terminal window.
2. Execute the createdb.db2.sh script to create the WCS database as follows:

```
# su - db2inst1
$ cd /usr/lpp/CommerceSuite/bin
$ ./createdb.db2.sh wcs db2inst1 password
```

Syntax: createdb.db2.sh

```
> createdb.db2.sh <database_name> <db2_instance_owner>
<db2_instance_owner_passwod>
```

3. A logfile will be created called creatdb.log in the current directory. Review the contents of the createdb.log file created in the current directory to verify the database was created successfully.

4. Once the database is created, the database needs to be populated with the WCS schema.
5. Execute the populatedb.db2.sh to populate the WCS database with the WCS schema is as follows:

```
# su - db2inst1
$ cd /usr/lpp/CommerceSuite/bin
$ ./populatedb.db2.sh wcs db2inst1 password
```

Syntax: populatedb.db2.sh

```
> populate.db2.sh <database_name> <db2_instance_owner>
<db2_instance_owner_passwod>
```

6. Execute the populatedbnl.sh as follows:

```
# su - db2inst1
$ cd /usr/lpp/CommerceSuite/bin
# populatedbnl.sh wcs db2inst1 <db2_instance_owner_password>
```

Syntax: populatedbnl.db2.sh

> populatedbnl.db2.sh <database_name> <db2_instance_owner>
<db2_instance_owner_passwod>

7. Catalog the WCS database on the Commerce/WebSphere Application Server (machine B).

   Once these three scripts have been executed successfully, the WCS database creation is complete. Prior to the WCS instance being created, the WCS database must be cataloged as follows:

```
# su - db2inst1
# db2 connect to riscwas1 user db2inst1 using <db2_instance_owner_password>
# db2 catalog db wcs at node pluto
```

The WCS database has been created and is now ready for the WCS instance creation.

7.8.12 Step 12: Create a WCS instance

Provided the WCS database has been created manually as described in 7.8.11, “Step 11: Create the WCS database manually” on page 190, the WCS instance can be created by using the Configuration Manager.

To create a WCS instance using the Configuration Manager for a remote database, complete the following steps:

1. This section details the steps to take when creating a WCS instance via the “Configuration Manager” with a database on a separate machine. From earlier steps, we created the database manually on the Database server, so we have to prevent the GUI from creating it. To do this, place comments in the following files:

   – createdb.db2.sh

   – populatedb.db2.sh

   – populatedbnl.sh

   This is done by placing a comment character at the beginning of each line in the shell/script files. The UNIX shell command’s comment character is “#”.

   The createdb.db2.sh file should look like this:

```
#!/bin/ksh

. $PWD/setenv.sh

DATABASE=$1
USER=$2
PASSWORD=$3

#SCHEMA=$WCS_HOME/schema
LOG=createdb.log

#db2 create database $DATABASE using codeset UTF-8 territory US > $LOG
#db2 update database configuration for $DATABASE using applheapsz 16384 >>
$LOG
#db2 update database configuration for $DATABASE using stmtheap 8192 >>
$LOG
#db2 update database configuration for $DATABASE using app_ctl_heap_sz 512
>> $LOG
#db2 update database configuration for $DATABASE using locklist 400 >> $LOG
#db2set DB2_RR_TO_RS=yes >> $LOG
#
db2 connect to $DATABASE user $USER using $PASSWORD >> $LOG
#db2 -tvf $SCHEMA/db2/wcs.schema.sql >> $LOG
#db2 -tvf $SCHEMA/db2/wcs.key.sql >> $LOG
#db2 -tvf $SCHEMA/db2/wcs.index.sql >> $LOG
#db2 -tvf $SCHEMA/db2/wcs.view.sql >> $LOG
#db2 -tvf $SCHEMA/db2/migration/5100_5101.sql >> $LOG
db2 terminate >> $LOG
The popoulatedb.db2.sh file should look like:
#!/bin/ksh
. $PWD/setenv.sh
DATABASE=$1
USER=$2
PASSWORD=$3
```

```
DBPARMS="-dbname $DATABASE -dbuser $USER -dbpwd $PASSWORD"

LOG=populatedb.log
#SCHEMA=$WCS_HOME/schema
#XML=$WCS_HOME/schema
#LIB=$WCS_HOME/lib/loader
#MASSLOADER=com.ibm.wca.MassLoader.MassLoad
#
#CP1=$LIB/DTDGenerator.zip:$LIB/IdResGen.zip:$LIB/Logger.zip:$LIB/MassLoade
r.zip
#CP2=$LIB/SAFServ.zip:$LIB/WCALogger.zip:$LIB/WCMCommon.zip:$LIB/WCSTasks.z
ip
#CP3=$LIB/jgl2.0.0.jar:$WCS_HOME/lib/jlog.jar:$WCS_HOME/xml/loader:$LIB/wcm
xmlp.jar:$LIB/wcmxslt.jar
#
#export
CLASSPATH=$CP1:$CP2:$CP3:$LIB/db2/dbconnect.zip:$DB2_DRIVER:$CLASSPATH
#export LIBPATH=$LIBPATH:$DB2_HOME/lib
#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DB2_HOME/lib
#
#if [ "$OS_NAME" = "AIX" ] ; then
#        if [ -f /usr/bin/java ]; then
#                TEMP=`(/usr/bin/java -fullversion 2>&1 | /usr/bin/grep
"1.1.8")`
#                if [ $? -eq 0 ]; then
#                        JAVA_EXE=/usr/bin/java
#                        export LIBPATH=/usr/jdk_base/lib/aix:$LIBPATH
#                        export
CLASSPATH=/usr/jdk_base/lib/classes.zip:$CLASSPATH
#                fi
#        fi
#fi
#
#cd $XML
#
The populatedb.dbnl.db2.sh file should look like this:
!/bin/ksh
. $PWD/setenv.sh
DATABASE=$1
USER=$2
PASSWORD=$3
DBPARMS="-dbname $DATABASE -dbuser $USER -dbpwd $PASSWORD"
LOG=populatedbnl.log
#SCHEMA=$WCS_HOME/schema
#XML=$WCS_HOME/schema
#LIB=$WCS_HOME/lib/loader
#MASSLOADER=com.ibm.wca.MassLoader.MassLoad
#
```

```
#CP1=$LIB/DTDGenerator.zip:$LIB/IdResGen.zip:$LIB/Logger.zip:$LIB/MassLoade
r.zip
#CP2=$LIB/SAFServ.zip:$LIB/WCALogger.zip:$LIB/WCMCommon.zip:$LIB/WCSTasks.z
ip
#CP3=$LIB/jgl2.0.0.jar:$WCS_HOME/lib/jlog.jar:$WCS_HOME/xml/loader:$LIB/wcm
xmlp.jar:$LIB/wcmxslt.jar
#
#export
CLASSPATH=$CP1:$CP2:$CP3:$LIB/db2/dbconnect.zip:$DB2_DRIVER:$CLASSPATH
#
#if [ "$OS_NAME" = "AIX" ] ; then
#        if [ -f /usr/bin/java ]; then
#                TEMP=`(/usr/bin/java -fullversion 2>&1 | /usr/bin/grep
"1.1.8")`
#                if [ $? -eq 0 ]; then
#                        JAVA_EXE=/usr/bin/java
#                        export LIBPATH=/usr/jdk_base/lib/aix:$LIBPATH
#                        export
CLASSPATH=/usr/jdk_base/lib/classes.zip:$CLASSPATH
#                fi
#        fi
#fi
#
#cd $XML

# IMPORT_LANG
```

On the Commerce/WebSphere Application Server (machine B), we now invoke the graphical user interface to complete the instance creation. The GUI will load the WAS repository with all the WCS relevant components (EJBs, etc.). The commands to start the GUI on the WAS machine can be found in the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.

Note: The WebSphere Application Server must be running prior to creating the instance through the GUI. The command to start the WAS is as follows:

```
# cd /usr/IBMWebAS/bin
# ./startupServer.sh &
```

2. Ensure the following servers are started:
   - Start the IBM HTTP Server:

     ```
     # cd /usr/HTTPServer/bin
     # ./apachectl start
     ```

   - Start the IBM DB2 Server:

     ```
     # su - db2inst1
     ```

```
$ db2start
```

– Start the IBM WebSphere Admin Server:

```
# cd /usr/IBMWebAS/bin
# ./startupServer.sh
```

3. Start the Configuration Manager Server as follows:

```
# cd /usr/lpp/CommerceSuite/bin
# ./config_server.sh
```

Note: Do not execute this command as a background process (&) to avoid a security risk.

4. Start the Configuration Manager (client) as follows:

```
# cd /usr/lpp/CommerceSuite/bin
# ./config_client.sh
```

5. When the Config Authentication window appears, enter the following and then click **OK**:

– User ID: `webadmin` (default)

– Password: `webibm` (default)

Note: The first time you log in, you will be prompted with the message `For security purposes, please change your default CM password.`

The Configuration Manager should now be displayed.

In the remaining steps, we provide examples for our runtime environment settings. Refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX* for all the possible configuration options and values.

6. Select and expand your `<hostname>` from the Configuration Manager window.
7. Select **Instance List**. Right-click **Instances** -> **Create Instance**.
8. When the Select an Option window appears with the message `Do you want to use an existing instance as the basis for this instance?`, click **No**.

Note: The Instance Creation Wizard window should appear. This wizard will guide you sequentially through each tab to configure components. You can go back to any of these tabs directly by clicking on it.

9. Select the **Instance** tab (default starting point), and enter the following and then click **Next**:

– Instance name: `wcs`

– Instance root path: `/usr/lpp/CommerceSuite/instances/wcs`

Note: You must manually change the instance name. By default the directory will be demo.

– Deselect the **Merchant Key** checkbox.

– Merchant Key: `<your_merchant_key>`

The merchant key requires a 16-character hexadecimal value. This is a user-defined field.

Note: To avoid a security risk, you must enter your own defined value for the merchant key.

10. On the Database tab, enter the following and then click **Next**:

– Database administrator name: `db2inst1`

– Database administrator password: `<db2inst1_password>`

– Database name: `wcs`

– Database type: select `DB2` from the pull-down

– Database user name: `db2inst1`

– Database user password: `<db2inst1_password>`

– Check **Set as active database** (default)

– Check **Use remote database**

Note: The Use remote database option must be checked if you are using a remote database (for example, a 3-tier configuration).

11. On the Languages tab, accept the default (wcs.bootstrap_multi_en.xml), and click **Next**.

12. On the Web Server tab, enter the following and then click **Next**:

– Hostname: `jganci2.itso.ral.ibm.com`

It should detect this automatically. If not, enter the fully qualified hostname.

– Web Server Type: select **IBM HTTP Server** from the pull-down

– Primary Document Root: `/usr/HTTPServer/htdocs/en_US`

– Server Port: `80`

13. On the WebSphere tab, enter the following and then click **Next**:

– Data Source Name: `WebSphere Commerce Suite DB2 DataSource wcs`

The wcs instance name is added at end of this string to create the data source name.

- Port Number: 900 (default)
- JDBC Driver Location: /usr/lpp/db2_07_01/java12/db2java.zip
- Check Stores Web Application (default)
- Check Tools Web Application (default)

14. On the Payment Manager tab, accept the defaults, and click **Next**.

    In this example, we do not configure Payment Manager.

15. On the Log System tab, accept the defaults, and click **Next**.
16. On the Messaging tab, accept the defaults, and click **Finish**.
17. When the Population Confirmation window appears, click **Yes**.

> **Note:** The instance creation process takes approximately 10-20 minutes depending on your system specifications.

18. When the Configuration Manager is done creating the instance, you should see a message `Instance was successfully created`. Click **OK** and close the WCS Configuration Manager.

> **Note:** We recommend that you stop the Configuration Manager Server after the WCS instance is created to avoid a security risk and reduce memory consumption.

19. Restart the WebSphere Commerce Server.

    After the WCS instance is created, the WebSphere Commerce Server - <wcs_instance> (application server), must be restarted by completing the following steps:

    a. Start the WebSphere Administrator's Console as follows:

    ```
    # cd /usr/IBMWebAS/bin
    # ./adminclient.sh
    ```

    b. Select and expand **WebSphere Administrative Domain** -> **<your_host>**.

    c. Select **WebSphere Commerce Server - wcs**. Right-click **Stop**. Once the server has stopped, Right-click **Start**.

20. Verify the WCS instance.

    The Configuration Manager WCS instance creation is now complete. Proceed to 7.6.3, "Verifying a WCS instance" on page 172.

Important:

1. The Instance's root path is correct. As the configuration wizard does not change the instance's root path when you enter the instance name, the root path must be set correctly manually.
2. The instance file, `<instance_name>.xml`, must contain the value "-1" in the Supported Languages field of the file. If this value is not present, the Store Services page will display an empty body page to a browser.

When the Configuration Wizard has successfully completed, we need to modify the entry for the Cleanup Agent Hostname setting for the instance. We do this because in a 3-tier environment we have an HTTP Server separate from that of our Commerce/WebSphere Application Server. To complete this task, do the following:

1. On the Configuration Manager (not the Configuration Wizard) expand the instances list by clicking the + sign next to the word Instances.
2. Expand the instances list by clicking the + sign next to your instance name.
3. Click the word **Cache** within your instance hierarchy display. This will produce a dialog box on the right-hand side frame titled cache - <instance_name>.
4. Scroll down this pane to show the Cleanup agent hostname. This will say localhost. Change this to the name of your remote HTTP Server machine name.
5. Click **Apply** to save the changes.

7.8.13 Step 13: Add the WCS entries to the httpd.conf

During the WCS instance creation in a 3-tier environment, the WebSphere Application Server is unable to make the appropriate changes to the remote HTTP Server configuration file (httpd.conf). These changes have to be made manually.

1. Log on to the HTTP Server as root and create a terminal window.
2. From within the terminal window, stop the Web server, by issuing the following command:

   ```
   /usr/HTTPServer/bin/apachectl stop
   ```

3. Edit the /usr/HTTPServer/conf/httpd.conf file and add the following lines:

   ```
   <directory "/usr/HTTPServer/htdocs/en_US">
   Options Indexes
   AllowOverride None
   order allow,deny
   allow from all
   ```

```
</Directory>
<Directory “/usr/lpp/CommerceSuite/web”>
<Files *.jsp>
order allow,deny
deny from all
</Files>
</Directory>
Alias /accelerator
“/usr/lpp/CommerceSuite/web/tools/common/accelerator.html”
Alias /storeservices
“/usr/lpp/CommerceSuite/web/tools/devtools/storeservices.html”
Alias /adminconsole “/usr/lpp/CommerceSuite/common/wcsadmincon.html”
Alias /wcs “/usr/lpp/CommerceSuite/web”
Alias /wcshelp “/usr/lpp/CommerceSuite/web/doc/en_US”

<VirtualHost your_HTTP_server_IP_address>
ServerName your_HTTP_server_FQDN
DocumentRoot “/usr/HTTPServer/htdocs/en_US”
</VirtualHost>

<VirtualHost your_HTTP_server_IP_address:443>
SSLEnable
SSLClientAuth 0
ServerName your_HTTP_server_FQDN
DocumentRoot “/usr/HTTPServer/htdocs/en_US”
</VirtualHost>
```

4. Once these changes have been made, the HTTP Server needs to be re-started.

```
# cd /usr/HTTPServer/bin
# ./apachectl start
```

At this point, the HTTP Server is now configured to accept the Web requests for the commerce store and pass them to the Commerce/WebSphere Application Server.

7.8.14 Step 14: Deploy the sample store

This section documents the process of publishing a store once the development of that store has completed. This section assumes that the store developers have completed development and have delivered a store archive file - SAR file for deployment in the production/staging environment.

A summary of the steps documented in this section are as follows:

1. Prepare WCS for new SAR file
2. Store deployment via graphical user interface

3. Store deployment via command line
4. Testing of the deployed store

As expected, store deployment on a single-tier is straightforward. The deployment on the two-tier environment is also straightforward, since the Web server and application server are sharing infrastructure.The deployment of a 3-tier environment is more complex, because it involves the separation of the store's static components, generally HTML files and images, from the store logic components, such as application beans, enterprise beans XML files, and JSP files.

Deploy the sample store using Store Services

This section outlines the steps involved to publish a WCS store using the graphical user interface (GUI) called Store Services.

Prepare WCS for new SAR file

This section details the steps involved to prepare the WebSphere Commerce Suite software environment for the new SAR (store archive) file. These steps are independent of the AIX runtime topology, and are required when publishing via the Store Services.

1. Log on to the WCS/WAS server as root and create a terminal window.
2. In this terminal window, create a directory for the new store:

   ```
   # mkdir /usr/lpp/CommerceSuite/samples/stores/<storename>
   ```

   where <storename> is the name of the store being deployed
3. Copy the SAR file into the directory that has been created:

   ```
   /usr/lpp/CommerceSuite/samples/stores/<storename>/storename.sar
   ```
4. For the GUI to be able to see the new SAR file, modify the file SARRegistry.xml file to include the new SAR file. An example of this is:

   ```
   <SampleSAR fileName="<storename.sar>" relativePath="<storename>"
          <html locale="en_US" featureFile="RetailModel/Feature_en_US.html"
             sampleSite="RetailModel/Preview/en_US/index.html"/>
          <html locale="fr_FR" featureFile="RetailModel/Feature_fr_FR.html"
             sampleSite="RetailMOdel/preview/fr_FR/index.html"/>
   </SampleSAR>
   ```

   In this example, we are going to publish an English and a French language store.

Once these steps are completed, the WCS Store Services Web page will be able to see this new SAR file.

Publish the store from Store Services

This section documents how to use WCS Store Services to publish the new store.

1. From a Web browser, launch the Store Services page.

   `http://<http_server_machine>/storeservices`

   Note: In a 3-tier environment the URL is the `http://<http_server_machine>/storeservices`. In a 1-tier and 2-tier environment, the URL is `http://<wcs/was_server_machine>/storeservices`.

2. Log on using the username of wcsadmin and enter your wcsadmin password. This generates a new browser window that Store Services operates. The browser presents a list of existing SAR files that are already created from previous store publishing activities.
3. Click **New** to create a new publishing SAR file.
4. The next window shows a list of SAR files to choose from. The new SAR will be listed here. Select the new SAR file.
5. Enter the appropriate information for the Store Archive and Store Directory. In this example we chose **proto2** for both values.
6. Click the **OK** button.
7. A message box will appear indicating that the same file was created successfully. Click the **OK** button to dismiss the message box. You are returned to the Store Services home page, which has the new SAR file listed.
8. Select the newly created SAR file, then click the **Publish** button. This will publish the contents of the SAR file to the WCS directory structure. A confirmation window will be displayed; click **OK** to continue.
9. The Store Services home page is then displayed with the status of the new store as Publishing. This will take some time, depending on the amount of contents to be published. You can click **Refresh** to see any changes in the store's status.

   Once the message `Publishing Successfully Completed` is displayed next to the store name, the publishing stage has finished, and you can now launch the store.
10. On the Store Services home page, select the store archive we have just published and click **Publish Summary**.
11. This will present the Publish Summary page. Click **Launch Store** to view the store. This presents a dialog box asking for the Web application path for the store. Accept the default and click **OK** to continue. This launches the store in a new browser window.

Testing sample store

Once the store has been published, you can test the functionality of the store by entering the URL of your store. For example:

```
http://<store_name>/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=<
store_id>&langId=-1&catalogId=<catlog_id>
```

Where <store_name> is the name of the store being published, <store_id> is the store_id value corresponding to the new store, and <catalog_id> is the catalog-id corresponding to the new catalog.

The store front should now appear in the browser window to confirm the store has been deployed.

7.8.15 Step 15: Serve static content from the Web server

This section describes the procedure to move the static content (HTML, images, etc.) from the WebSphere Application Server to the HTTP Server. There are a number of ways of sharing files with the Commerce/WebSphere Application Server and the HTTP Server. The example shown here increases the performance of the HTTP Server by having the HTML files stored on the HTTP Server itself, while the business logic supporting the store function is resident on the Commerce/WebSphere Application Server (machine B), which would be behind a corporate firewall, and therefore in a higher-trust environment.

1. Log on to the HTTP Server as user ID root, and start a terminal window.
2. Stop the Web Server as follows:

   ```
   # cd /usr/HTTPServer/bin
   # ./apachectl stop
   ```

3. Make a local copy of the /usr/lpp/CommerceSuite/web directory on the HTTP Server.

   ```
   # mkdir /usr/lpp/CommerceSuite
   # mkdir /usr/lpp/CommerceSuite/web
   ```

4. On the WCS/WAS server, tar the contents of the /usr/lpp/CommerceSuite/web so it can be copied to the HTTP Server:

   ```
   # cd /usr/lpp/CommerceSuite
   # tar cvf wcs_web.tar web
   ```

   Note: Be aware this is a large amount of data, so be sure that there is adequate space for the tar file.

5. Copy the tar file to the HTTP Server. This can be achieved by using either FTP or NFS.

6. Once copied, unpack the tar file to the /usr/lpp/CommerceSuite/web directory:

```
# cd /usr/lpp/CommerceSuite
# tar xvf wcs_web.tar
```

7. At this point, we have all the HTML and JSP pages on the external HTTP Server, which is not desirable. We now must remove the JSP pages, by deleting all *.jsp (JSPs) from the /usr/lpp/CommerceSuite/web directory structure.

At this point, we now have the static HTML files residing on the HTTP Server. The JSPs now reside on the Commerce/WebSphere Application Server (machine B).

7.8.16 Step 16: Remote administration considerations

The environments shown in this book are not typical of deployed production environments, since security and firewalls have not been used. The store administration in this book is not typical of a production environment, because the HTTP Server has been used as an entry point into the environment for store users as well as administrators. This section outlines some considerations for the remote administration of a multi-tier production environment.

Figure 7-1 shows a typical production WebSphere Commerce Suite environment showing the added security infrastructure. This shows the installation of additional equipment such as router, firewalls, etc.

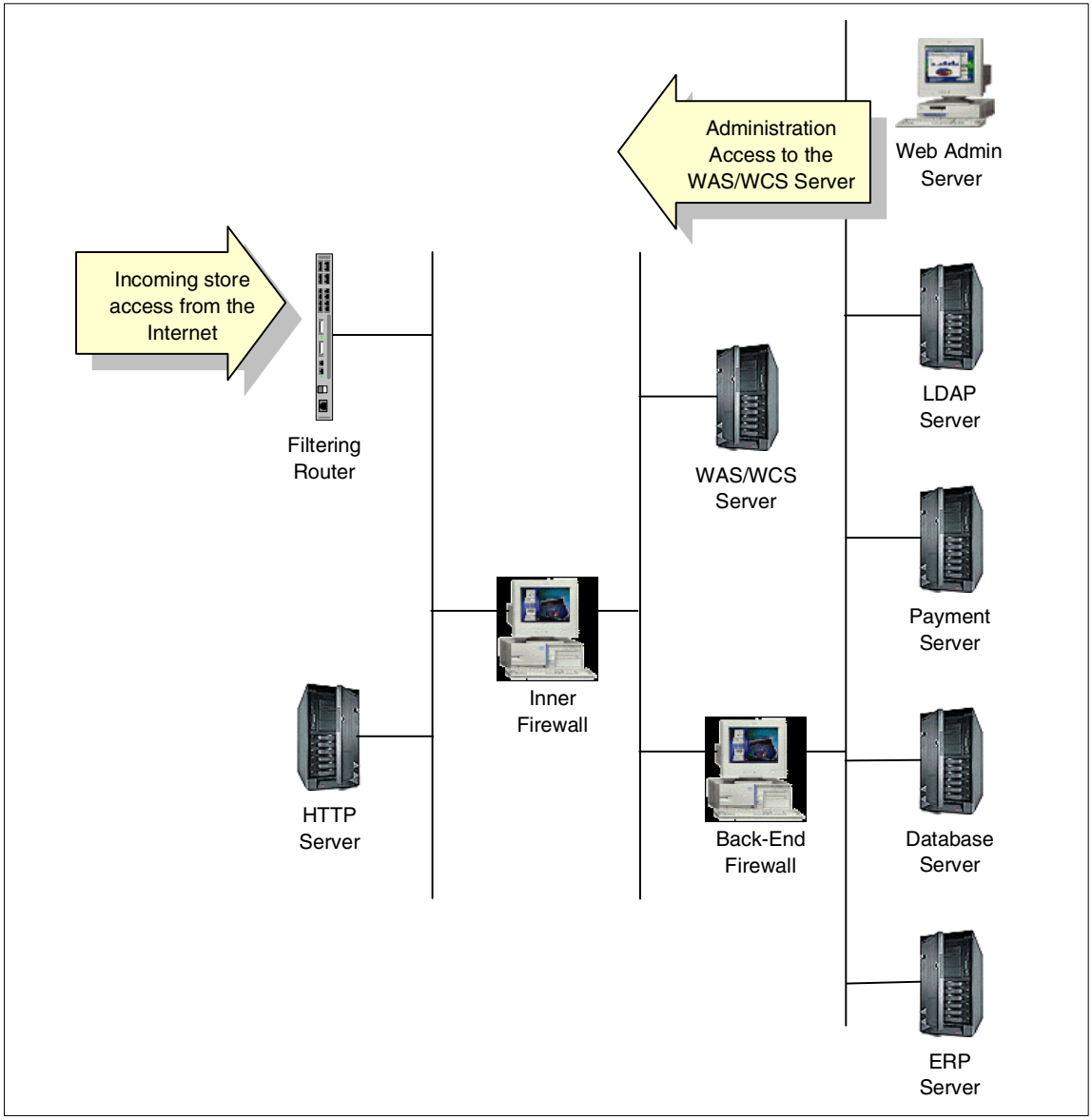


Figure 7-1 Sample 3-tier runtime environment

In this sample production environment, it is recommended that the administration of the environment not be conducted through the external HTTP Server, but from a HTTP Server within the secure back-end environment. This implementation restricts external/internet access to the administration components of the WebSphere Commerce Suite store, while allowing the store administration to be completed from a secure/trusted environment (that is, intra-company).

In order to achieve this environment, another HTTP Server would be required. This does not necessarily mean another hardware component, since the software component (HTTP Server) can reside on another machine, such as a database server or the payment server. In this example, it resides on a dedicated machine. The software steps required to complete the remote administration would be as follows:

1. Install the HTTP Server software as though the HTTP Server were the server providing HTTP support for the WCS environment. Follow the steps detailed in 7.2, “Install the IBM HTTP Server” on page 144.
2. At this stage the Web server is installed, and the OSERemote plug-in has been configured to send servlet requests to the WebSphere Application Server.

Note: There is no need to copy the store contents to this administration server. However, you will need to copy the HTML files that comprise the store administration components.

3. Ensure the changes are made to the WebSphere Application Server’s default_host configuration to accept connections from the internal HTTP Server’s hostname/address, and restart the Application Server.
4. Test the connection to the WebSphere Application Server, by entering the Store Services page address into a Web browser within the back-end network:

   ```
   http://<HTTP_Server>/storeservices
   ```

5. Remove the aliases defined in the external HTTP Server for the administration function of the store from the httpd.conf file. For example:

   ```
   Alias /accelerator
   “/usr/lpp/CommerceSuite/web/tools/common/accelerator.html”
   Alias /storeservices
   “/usr/lpp/CommerceSuite/web/tools/devtools/storeservices.html”
   Alias /adminconsole “/usr/lpp/CommerceSuite/common/wcsadmincon.html”
   Alias /wcs “/usr/lpp/CommerceSuite/web”
   Alias /wcshelp “/usr/lpp/CommerceSuite/web/doc/en_US”
   ```

6. Restart the external HTTP Server.

By following the guidelines above, the administration function of the WCS environment will be removed from the front-end of the store environment to the back-end environment. This will provide a more trusted store administration environment.

7.8.17 Step 17: Configure the IPSec tunnel

The OSE protocol does not include a security layer. To avoid a security risk, a security scheme must be implemented. In this example, we describe how to use IPSec tunneling to secure the OSE remote connection between the Web Server (machine A) and the Commerce/WebSphere Application Server (machine B).

Note: For more detailed information on IPSec tunneling, refer to *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855.

This section is organized into the following phases:

- Prerequisites for the IPSec tunnel
- Configuring the IPSec tunnel
- Starting the IPSec tunnel

Prerequisites for the IPSec tunnel

The IPSec tunnel software is provided as a component of the IBM AIX 4.3.3 operating system. The only prerequisites for configuring IPSec tunneling are the AIX 4.3.3 file sets listed in Table 7-9.

Table 7-9 AIX file sets required for IPSec tunnel

File set	Version installed	CD-ROM Number
bos.net.ipsec.rte	4.3.3.0 4.3.3.25	AIX 4.3.3 Vol 2 CD and AIX 4.3.3 Update CD-ROM
bos.msg.en_US.net.ipsec * where en_US is the locale	4.3.3.0 4.3.3.25	AIX 4.3.3 Vol 2 and AIX 4.3.3 Update CD-ROM
bos.net.ipsec.keymgt	4.3.3.0 4.3.3.25	AIX 4.3.3 Vol 2 CD and AIX 4.3.3 Update CD-ROM
bos.net.ipsec.websm	4.3.3.0 4.3.3.25	AIX 4.3.3 Vol 2 CD and AIX 4.3.3 Update CD-ROM
bos.crypto (for the specific country)	4.3.3.0 4.3.3.25	AIX 4.3.3 Bonus Pack CD-ROM

File set	Version installed	CD-ROM Number
bos.crypto-wt (for the specific country)	4.3.3.0 4.3.3.25	AIX 4.3.3 Bonus Pack CD-ROM
gskit.rte	4.3.3.0 4.3.3.25	AIX 4.3.3 Bonus Pack CD-ROM

Once these file sets are installed on the Commerce/WebSphere Application Server (machine B) and the Web Server (machine A), we are ready to configure the IPSec tunnel between them.

Configuring the IPSec tunnel

This section details the steps involved to configure the tunnel. The tunnel is configured on one end-point first, then exported to the other end-point. The steps below will complete this configuration. In our testing we started on the HTTP Server machine.

1. Log in to AIX as root and start a terminal session.
2. Once installed, we need to load the IP security stack. Use the `smitty` command to do this with the `ipsec4` shortcut.

   ```
   # smitty ipsec4
   ```

3. Select the option **Start/Stop IP security** and press Enter.
4. In the Start IPSecurity window, accept the defaults, which are:
   - Start ip security = now & after reboot
   - Deny all non-secure ip packets = no
5. Press Enter to start IP security.
6. Confirm the devices are available. This command will provide a list of IPSec devices that have started:

   ```
   # lsdev -Cc ipsec
   ```

7. Create the tunnel definition. In our testing we used a manual tunnel. Manual tunnels can be used between any two hosts that are running IP security and have a common set of cryptographic and authentication algorithms.

   ```
   # gentun -v 4 -t manual -s <source_ip_addr> -d <dest_ip_addr> -a HMAC_HD5
   -e DES_CBC_8 -N 23678
   ```

   Where `<source_ip_addr>` is the IP address of the HTTP Server machine, and `<dest_ip_addr>` is the IP address of the WCS/WAS machine.

   A message is displayed indicating the creation of tunnel 1 was successful.

Note: Do not start the tunnel yet. We need to export the definition file for this tunnel to the WCS machine. We cannot do this while the tunnel is active.

8. Export the tunnel definition file to the tunnel end-point, in this case the Commerce/WebSphere Application Server. The following command creates a tunnel definition file in the /tmp directory.

   ```
   # exptun -v 4 -t 1 -f /tmp
   ```

9. Copy the tunnel definition file, ipsec_tun_manu.exp to the other tunnel end-point server's /tmp directory, in our case the WCS/WAS server. During our testing we used FTP to transfer this file. This should be done in a more secure fashion, such as on a diskette.

10. On the other tunnel end-point, import the tunnel definition file.This will create a corresponding tunnel from the WCS/WAS server to the HTTP Server. Log on to the WCS/WAS server as root and create a terminal window. In this terminal window issue the following command to create the tunnel:

    ```
    # imptun -v 4 -t 1 -f /tmp
    ```

11. Confirm the tunnel is defined, by issuing the following command:

    ```
    # lntun -v 4
    ```

Starting the IPSec tunnel

Once the tunnel has been configured, all that remains is to start it. This command has to be issued on both tunnel end-points.

1. Log on the to HTTP Server as root and create a terminal window.
2. In this window, issue the following command

   ```
   # mktun -v -t 1
   ```

3. If you `ping` the Commerce/WebSphere Application Server, before starting the tunnel on the Commerce/WebSphere Application Server, it will fail. This is a simple demonstration that tunnel is active on the HTTP Server.
4. Log on to the WCS/WAS server as `root` and create a terminal window.
5. In this window, issue the following command:

   ```
   # mktun -v -t 1
   ```

6. At this time, both tunnel end-points are active. A ping from either end-point to the other end-point will be successful.

7.9 Where to find more information

- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX* found on the WebSphere Commerce Suite V5.1 CD.
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *WebSphere Edge Server: Working with Web Traffic Express and Net Dispatcher*, SG24-6172
- *A Secure Way to Protect Your Network: IBM SecureWay Firewall for AIX V4.1*, SG24-5855
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153

8

WCS runtime for Solaris: Oracle8i and iPlanet

This chapter provides detailed procedures for installing, configuring, verifying, and deploying WebSphere Commerce Suite V5.1, Pro Edition for Solaris in a 2-tier configuration using Oracle8i Enterprise Edition and iPlanet Web Server.

These procedures are intended to be used as a working examples in conjunction with the product installation guides, which provide information on all the possible values that may be unique for your runtime environment.

This chapter is organized into the following sections:

- WCS runtime environment for Solaris
- Install the iPlanet Web Server
- Install the Oracle8i Server
- Install the Oracle8i Client
- Install the WebSphere Application Server
- Install the WebSphere Commerce Suite
- Create a WCS instance
- Deploy the sample store
- Where to find more information

© Copyright IBM Corp. 2001

8.1 WCS runtime environment for Solaris

This section defines the hardware and software used within our test WCS runtime environment for Solaris. Our example WCS runtime environment for Solaris is a 2-tier configuration that uses an Oracle8i database server and the iPlanet Web Server.

8.1.1 Hardware and software prerequisites

For detailed information on hardware and software prerequisites, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*.

In addition, when setting up a 2-tier configuration we recommend that you refer to the *Oracle8i Installation Guide*, Release 2 (8.1.6) for Sun SPARC Solaris, Part No. A77181-01, for hardware and software prerequisites on the Oracle8i Enterprise Edition database server.

8.1.2 Hardware and software used for 2-tier configuration

This section describes the hardware and software used within our WCS runtime environment for a Solaris 2-tier configuration.

Machine A - WCS Server

We used the following hardware:

- Sun SPARC Ultra 60 (600-6486-01)
  - 450 MHz UltraSPARC II CPU
  - 1 GB RAM
  - 17 GB hard disk
  - 1 Ethernet adapter (built-in)
  - Hostname: sun1.itso.ral.ibm.com

We installed the following software on this system:

- Solaris 7 and patches required for WCS V5.1
- Netscape iPlanet Web Server V4.17
- Oracle8i Enterprise Edition, R2 (8.16) Database Client
- IBM WebSphere Application Server V3.5, Advanced Edition
- WebSphere Application Server V3.5 Fixpack 2 + E-fixes for WCS

- WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 Pro for Solaris

Machine B - Oracle8i Database Server

We used the following hardware:

- Sun SPARC Ultra 60 (600-6486-01)
  - 450 MHz UltraSPARC II CPU
  - 1 GB RAM
  - 17 GB hard disk
  - 1 Ethernet adapter (built-in)
  - Hostname: sun2.itso.ral.ibm.com

We installed the following software on this system:

- Solaris 7 and patches required for WCS V5.1
- Oracle8i Enterprise Edition, R2 (8.16) Database Server

8.1.3 Install Solaris 7 and patches required for WCS V5.1

For detailed instructions on installing the Solaris 7 operating system and patches required for WebSphere Commerce Suite V5.1, Pro Edition for Solaris on SPARC systems refer to“Solaris 7 installation” on page 586.

8.2 Install the iPlanet Web Server

This section provides detailed instructions for installing, configuring, and verifying the Netscape iPlanet Web Server, Enterprise Edition V4.1 SP7 (4.17).

The iPlanet Web Server installation is organized into the following phases:

- Install Netscape Communicator
- Pre-installation steps for the iPlanet Web Server
- Install the iPlanet Web Server
- Enable SSL for iPlanet Web Server
- Verify iPlanet Web Server

Note: More detailed information on iPlanet Web Server can be found at:

http://developer.iplanet.com/docs/manuals/enterprise.html

8.2.1 Install Netscape Communicator

The iPlanet Web Server requires Netscape Communicator V4.61 or higher (used for administration of tools).

Complete the following steps to install Netscape Communicator V4.76:

1. Start a Solaris Console, and enter the following:

   ```
   # cd /opt
   # mkdir ns476
   ```

2. Netscape Communicator V4.76 can be downloaded from the following URL:

   http://www.sun.com/solaris/netscape/getnetscape476.html

   Download the file to the /opt/ns476 directory.

   Note: Netscape Communicator V4.76 installation instructions can be found at http://www.sun.com/software/solaris/netscape/476_install.html

3. Uncompress the download file by typing one of the following commands:

   Check the name of the downloaded Netscape Communicator file.

   – If the Netscape file does not have a .Z suffix, your browser has uncompressed the Netscape file automatically. Use the following command to extract its files:

     ```
     # tar -xf <filename>
     ```

   – If the Netscape file has a .Z suffix, your browser did not uncompress the Netscape file automatically. Use the following command to uncompress and extract its files.

     ```
     # zcat <filename>.Z | tar -xvf -
     ```

4. Change to the directory where the compressed version of Netscape Communicator was extracted, for example:

   ```
   # cd /opt/ns476/NSCPcom
   ```

5. Install Netscape Communicator using the pkgadd utility, as follows:

   ```
   # pkgadd -d `pwd` NSCPcom
   ```

6. Follow the install instruction prompts.

7. To start Netscape Communicator do one of the following:

   – Command line:

     ```
     # cd /opt/NSCPcom
     # ./netscape
     ```

   – Click **Internet** (globe icon) from the Solaris 7 CDE Front Panel.

8. Add an entry to your PATH for the directory in which Netscape Communicator is installed. For example, if Netscape Communicator was installed in the default directory, add /opt/NSCPcom to the PATH. The path may be set in the .dtprofile, .login, or .cshrc file.

   For example, we added the following path to the /etc/.dtprofile file:

   ```
   export PATH=$PATH:/opt/NSCPcom
   ```

9. Configure Netscape Communicator by clicking **Edit** -> **Preferences** from the menu bar. We configured the SOCKS server, fonts, and home page.

8.2.2 Pre-installation steps for the iPlanet Web Server

Complete the following steps, before installing the iPlanet Web Server:

1. Ensure the server hostname and IP address are configured.

   For example:

   Host name: `sun1.itso.ral.ibm.com`

   IP Address: `9.24.105.46`

2. Determine if the server needs a DNS alias.

   You may wish to establish a DNS alias that points to the server machine. If a DNS alias will be needed, create it. In our example, we do not create a DNS alias.

3. Ensure that you have two unique port numbers available, one for standard Web access, and one for secure Web access.

   The most common ports are 80 (HTTP) for standard access and 443 (HTTPS) for secure access.

4. Create a UNIX user account for the iPlanet Web Server. This account should have restricted access for security reasons.

   ```
   # groupadd iplanet
   # useradd -g iplanet -d /export/home/iplanet -m -s /usr/bin/ksh iplanet
   # passwd iplanet
   ```

   After the last command, you will be prompted for a password for the new user, and then asked to confirm it.

8.2.3 Install the iPlanet Web Server

To install the iPlanet Web Server, complete the following steps:

1. Start the Solaris Console and log in as root.
2. Insert the iPlanet Web Server CD-ROM in the WCS Server system CD-ROM drive.

3. Copy the iPlanet Web Server tar file to a temp directory on your system.

   For example:

   ```
   # mkdir -p /opt/ip417
   # cd /<root_iplanet_cdrom>/solaris/enterprise
   # cp enterprise.tar /opt/ip417/
   ```

4. Uncompress the tar file by typing the following:

   ```
   # cd /opt/ip417
   # tar -xvf enterprise.tar
   ```

5. Start the iPlanet Web Server install by typing the following:

   ```
   # ./setup
   ```

6. When you are prompted with the message `Would you like to continue with the installation [Yes]:` press Enter (for Yes as default).

   Note: The following tips are used for navigation during the install.

   - Press Enter to choose the default and go to the next window.
   - Press Ctrl+B to back to the previous window
   - Press Ctrl+C to cancel the installation program
   - You can enter multiple items using commas to separate them, for example 1,2,3.

7. When you are prompted with the message `Do you agree to license terms?` type `Yes`, and then press Enter.
8. When you are prompted with the message `Choose the installation type [2]:` press Enter to accept the default (Typical installation).
9. When you are prompted with the message `Install location [/usr/netscape/server4]:` press Enter to accept the default.
10. When you are prompted with the message `Specify the components you wish to install [All]:` press Enter to accept the default.
11. When you are prompted with the message `Specify the components you wish to install [1,2,3,4,5,6,8]: type 1,2,3,4,5` and then press Enter.

Note: The following components are required by WebSphere Commerce Suite:

1 Server Core

2 Java Runtime Environment

3 Java Support

4 SSJS Support

5 SSJS Database Support

12. When you are prompted with the message `Computer name [your_hostname.domain]:` press Enter.
13. When you are prompted with the message `System User [nobody]:` type the user created in the pre-installation steps, for example `iplanet`, and then press Enter.
14. When you are prompted with the message `System Group [nobody]:` type the group created in the pre-installation steps, for example `iplanet`, and then press Enter.
15. When you are prompted with the message `Run iWS Administration Server as [root]:` press Enter.
16. When you are prompted with the message `iWS Administration Server User Name [admin]:` press Enter. You will be prompted for a password.
17. When you are prompted with the message `iWS Admin Server Port [8888]:` press Enter.
18. When you are prompted with the message `Web Server Port [80]:` press Enter.
19. When you are prompted with the message `Do want to register this with an existing Directory Server [No]:` press Enter.
20. When you are prompted with the message `Web Server Content Root [/usr/netscape/server4/docs]:` press Enter.
21. When you are prompted with the message `Do you want to use your own JDK [No]:` press Enter.
22. Review messages to make sure everything extracted and installed successfully. Press Enter to continue and return to the command prompt.

The iPlanet Web Server installation is complete.

8.2.4 Enable SSL for iPlanet Web Server

This section provides detailed instructions for requesting a certificate, installing the certificate and configuring an iPlanet Web Server for SSL. In our example, we will request a test certificate from VeriSign. For a production environment you will need to request a real certificate from VeriSign.

The iPlanet Web Server SSL enablement is organized into the following sections:

- Create a Server for SSL transactions
- Create a certificate trust database
- Request a server certificate
- Install the server certificate
- Turn SSL encryption on

To enable SSL for iPlanet Web Server, complete the following steps:

1. Start the iPlanet Web Server Administration Server by typing the following commands:

```
# cd /usr/netscape/server4
# ./startconsole
```

2. When the Login window appears, enter the following:
   - User ID: `admin` (iWS Administration Server User Name created during the installation steps above)
   - Password: `<your_password>`

Create a Server for SSL transactions

3. If the Servers tab is not already selected, click the **Servers** tab.
4. Click **Add Server** in the left navigation pane.
5. In the Add Server window, enter the information for your SSL server as follows:
   a. In the Server Name field, enter your fully qualified host name, for example `sun1.itso.ral.ibm.com`.
   b. In the Server Port field, enter the port you chose as your SSL port, For example, `443`.
   c. In the Server Identifier field, enter a name which will identify your SSL server, for example `SSL`.
   d. In the Server User field, enter the UNIX user that you created for the Web server, for example `iplanet`.
   e. In the MTA Host field, accept the default of `localhost`.

f. Select **Never attempt to resolve IP addresses into host names** (the default).

g. In the Document Root field, accept the default, which will be the document root that you specified during installation, for example `/usr/netscape/server4/docs`.

h. Click **OK** to create the server.

Create a certificate trust database

6. In the **Success!** window, you will be shown a link to **Configure your new server**. Click this link.

> **Note:** You can configure your server at anytime from the Manage Server page of the iPlanet Administration Server by completing the following steps:
>
> - Enter URL: `http://<host>:8888/https-admserv/bin/index`.
> - In the Select a Server pull-down select the server you want to configure, for example SSL (Server name added for SSL in previous steps).
> - Click **Manage**.

7. On the Server Manager page, click the **Security** tab to create a Certificate Trust database.
8. Enter and re-enter a password to use for this database, then click **OK** to create the database.
9. A pop-up window should appear, indicating that a database was successfully created.

Request a server certificate

10. The next step is to generate a certificate request. Click the **Request a Certificate** link in the left navigation pane.
11. Generate your request as follows:

    a. Select the **New Certificate** radio button.

    b. In the CA-email address field, enter your e-mail address.

    c. Select **Internal (software)** from the Cryptographic module pull-down.

    d. Enter your certificate trust database password in the Key pair file password field.

    e. In the Common Name field, enter the URL by which your host will be known. This is normally the fully qualified host name, for example `sun1.itso.ral.ibm.com`.

f. In the remaining fields, enter your contact information.

g. Click **OK** to generate the certificate request.

12. The next window will inform you that your request has been sent. Take note of the file name specified by the message, `A copy of the certificate request has been saved in the file: <file_name>.`

13. Proceed to a Certificate authority Web site to submit your request. Follow the instructions provided there for the completion of your request.

Most certificate sites will allow you to generate a test certificate for use on non-production servers. For example, VeriSign can be found at:

`http://digitalid.verisign.com/server/index.html`

Install the server certificate

14. When you receive the certificate, go to the **Security** section of the server manager for the SSL server (if you are not already there), by doing the following:

    a. Enter the URL: `http://<host>:8888/https-admserv/bin/index`.

    b. In the Select a Server pull-down, select the server you want to configure, for example SSL (Server name added for SSL in previous steps).

    c. Click **Manage**.

    d. Click the **Security** tab.

15. Click the **Install Certificate** to install your new certificate.

16. Fill in the fields with the following values:

    a. In the Certificate For field, select **This Server**.

    b. In the Cryptographic module pull-down, select **Internal (software)**.

    c. In the Key Pair File Password field, enter the certificate trust database password defined above.

    d. Leave the Certificate Name field blank.

    e. Select the **Message Text** radio button.

    f. Paste the certificate sent to you by the certificate authority into the Message Text area.

    Note: Alternatively, you can save the certificate to a file, and specify the file name in the Message is in this file: text area.

    g. Click **OK** to begin installing the certificate.

17. In the Add Server Certificate page, take note of the information provided (for example, certificate name: Server-Cert) and click **Add Server Certificate**.

    A success window will pop up, as well as a window warning you that the server will need to be restarted for the changes to take effect.

Turn SSL encryption on

18. In order to activate SSL encryption, go to the Preferences section of the server manager for the SSL server (if you are not already there), by doing the following:
    a. Enter the URL: `http://<host>:8888/https-admserv/bin/index`.
    b. In the Select a Server pull-down, select the server you want to configure, for example SSL (server name added for SSL in previous steps).
    c. Click **Manage**.
    d. Click the **Preferences** tab.
19. Click **Encryption On/Off** in the left navigation pane.
20. Under Encryption, select the **On** radio button, verify the Port Number is `443` (or as previously defined), and then click **OK**.
21. When the warning message `Security changes require shutdown and restart` then click **OK**.
22. When the Save and Apply window appears, click **Save and Apply**.
23. When the Warning window message `The security information that you entered below will be sent in the clear, and may be exposed to network snooping` appears, enter the password for internal (software): `<trust_database_password>`.
24. A success window should appear. Click **OK**.
25. Both the HTTP and HTTPS servers will need to be restarted for changes to take effect.

8.2.5 Verify iPlanet Web Server

To verify that the iPlanet Web Server is working properly, complete the following steps:

1. Start the iPlanet Administration Server by entering the following URL:

   `http://<host>:8888/https-admserv/bin/index`.

2. In the Select a Server pull-down, select the HTTP Server, for example sun1.itso.ral.ibm.com.

Verify the HTTP server

3. Ensure that the HTTP Server is started. The server status will be displayed. If the server is not running, click **Server On** to start the server.

   If the server starts successfully you should see the message:

   ```
   The server is currently on.
   ```

4. Verify that the HTTP Server is running by accessing the server from a Web browser client, by entering the following URL: `http://<hostname>.`

Verify the HTTPS server

5. Ensure that the HTTPS Server (SSL) is started from the iPlanet Administration Server console. The server status can be checked by selecting the Select a Server pull-down for the SSL server. If the server is not running click **Server On** to start the server.
6. Verify that the HTTPS Server is running by accessing the server from a Web browser client, by entering the following URL: `https://<hostname>.`

The iPlanet Web Server install, configuration, and verification section is now complete.

8.3 Install the Oracle8i Server

This section provides detailed instructions for installing and configuring the Oracle8i Enterprise Edition Release 2 (8.16) for Sun SPARC Solaris on the Oracle8i database server system.

The Oracle8i Enterprise Edition database server install is organized into the following phases:

- Oracle8i install prerequisites
- Install Oracle8i Enterprise Edition Database Server
- Oracle8i post install configuration
- Oracle8i post install configuration
- Prepare Oracle8i databases for WAS and WCS

8.3.1 Oracle8i install prerequisites

Prior to the installation of Oracle8i you are required to complete the following prerequisites:

- Oracle8i disk space requirement

- Verify Solaris 7 patch levels required by Oracle8i
- Verify Solaris 7 required packages for Oracle8i
- Verify system executables
- Configure the UNIX kernel for Oracle8i
- Create UNIX groups for Oracle8i
- Create UNIX user account for Oracle8i
- Set environment variables
- Create mount points

Oracle8i disk space requirement

We recommend that you allocate the following disk space:

- Oracle8i Enterprise Edition Server 8.1.6

  Requires 1020 MB of free disk space.
- Oracle8i Client 8.1.6

  Requires 350 MB of free disk space.

The allocation of the free space required is detailed in “Solaris 7 installation” on page 586. We accounted for this when we installed Solaris 7 by allocating space to the /opt file system.

For details, refer to *Oracle8i Installation Guide, Release 2 (8.16) for Sun SPARC*, A77181-01.

Verify Solaris 7 patch levels required by Oracle8i

Prior to the installation of Oracle8i, we need to verify that the required level of Solaris 7 patches are applied.

1. Ensure the Solaris 7 patches are installed at the indicated level or higher as seen in Table 8-1.

Table 8-1 Solaris 7 patches required by Oracle8i (JRE 1.1.8_10)

Solaris 7 patch ID	Description	Required or Recommended
107636-01	X input and output method patch	Required
106980-05	Lib thread patch	Recommended
107607-01	Motif fontlist, fontset, libxm	Recommended
107078-10	Open Windows 3.6.1 Xsun patch (1)	Recommended

2. Verify the following Oracle8i (JRE 1.1.8_10) required patches are installed by typing the following command:

   ```
   # showrev -p | grep <patch_name>
   ```

   Where <patch_name> included without the level listed in Table 8-1.

   For example:

   ```
   # showrev -p | grep 107636
   ```

Verify Solaris 7 required packages for Oracle8i

Prior to the installation of Oracle8i, we need to verify that the Solaris 7 packages are installed.

1. Ensure the following Solaris 7 packages are installed:

Table 8-2 Solaris 7 packages required by Oracle8i

Solaris 7 package name	Description
SUNWarc	Archive libraries
SUNWbtool	CCS tools bundled with SunOS
SUNWhea	SunOS header files
SUNWlibm	Sun workshop bundled libm
SUNWlibms	Sun workshop bundled shared libm
SUNWsprot	Solaris bundled tools
SUNWtoo	Programming tools

2. Verify the Solaris 7 packages are installed by typing the following command:

   ```
   # pkginfo -i <package_name>
   ```

   Where <package_name> is among those listed in Table 8-2.

   For example:

   ```
   # pkginfo -i SUNWarc
   ```

Verify system executables

This step verifies that the correct version of the system executables is being used.

Type the following commands to verify system executables:

```
$ /usr/bin/which make
```

```
$ /usr/bin/which ar
$ /usr/bin/which ld
$ /usr/bin/which rm
```

The **which** command returns the executable path. The path search order is determined by the location of the path environment variable in the shell.

The required directory for the executable is /usr/ccs/bin. If this is not already in your path or not at the beginning of the path, you will need to add /usr/ccs/bin to the beginning of the path for the current shell.

Configure the UNIX kernel for Oracle8i

To ensure that Oracle8i has enough system memory, we need to configure the UNIX kernel.

To configure the UNIX kernel for Oracle8i, complete the following steps:

1. Start the Solaris Console and log in as user root.
2. Back up system file:

   ```
   # cd /etc
   # cp system system.bak
   ```

3. Add the following lines to the bottom of the system file:

   ```
   set shmsys:shminfo_shmmax=4294967295
   set shmsys:shminfo_shmmin=1
   set shmsys:shminfo_shmmni=100
   set shmsys:shminfo_shmseg=10
   set semsys:seminfo_semmni=100
   set semsys:seminfo_semmsl=100
   set semsys:seminfo_semmns=200
   set semsys:seminfo_semopm=100
   set semsys:seminfo_semvmx=32767
   ```

4. Type the following command to shut down and reboot:

   ```
   # ./usr/sbin/shutdown -i6 -g0 -y
   ```

Create UNIX groups for Oracle8i

Depending on the size and makeup of your organization, you may have several roles defined for database administration. The *Oracle8i Installation Guide, Release 2 (8.16) for Sun SPARC*, A77181-01, recommends that you create groups for the roles as seen in Table 8-3.

Table 8-3 Groups for Oracle8i

Role of Oracle8i group	Suggested group name
Database administrator	dba

Role of Oracle8i group	Suggested group name
Database operator	dboper
Oracle installer group	oinstall

In our example, we created groups called dba and oinstall as follows:

1. Start the Solaris Console and log in a root.
2. Create group by typing the following:

```
# groupadd dba
# groupadd oinstall
```

Create UNIX user account for Oracle8i

To create a user named `oracle` and make the user a member of the dba group, complete the following steps:

1. Start the Solaris Console and log in a root.
2. Create user oracle by typing the following:

```
# useradd -d /export/home/oracle -m -g oinstall -G dba -s /usr/bin/ksh
oracle
```

   Syntax:

```
> useradd -d <path_home_dir> -m -g <primary_group> -G <secondary_group> -s
<path_shell> <username>
```

   Where:

   - `-d <path_home_dir>` is the home directory of the user
   - `-m` creates the directory of the user and the .profile file
   - `-g` primary group to add user to
   - `-G` secondary group to add user to
   - `-s <path_shell>` is the path of the shell used by this user
   - `<username>` is the username to create

3. Create password for user oracle by typing the following:

```
# passwd oracle
```

   This will prompt the user for the password.

4. Verify login with user:

```
# su - oracle
$ su - oracle
```

   You will be prompted for your password.

Set environment variables

To set environment variables as part of the .profile file of the Oracle user shell, complete the following steps:

1. Log in as user oracle:

```
# su - oracle
```

2. Add the following environment variables to the oracle user profile:

```
umask 022
export ORACLE_BASE = /opt/oracle8/u01
export ORACLE_HOME = $ORACLE_BASE/app/oracle/product/8.1.6
export ORACLE_SID = <your_default_SID> (created in subsequent step)
export PATH=$PATH:$ORACLE_HOME/bin

# The following export is required for XWindows used by Oracle8i install.
export DISPLAY=sun2:0.0
```

Create mount points

To create mount points, complete the following steps:

1. Start the Solaris Console and log in a root.
2. Create the following directories:

   This directory is used for the Oracle8i software.

```
# mkdir -p /opt/oracle8/u01
```

   The following directories are intended to be used for databases:

```
# mkdir -p /opt/oracle8/u02
# mkdir -p /opt/oracle8/u03
# mkdir -p /opt/oracle8/u04
```

3. Change the owner of the directories to user oracle and group dba by typing the following command:

```
# chown -R oracle:dba /opt/oracle8
```

4. Change the permissions as follows:

```
# chmod -R g+w /opt/oracle8
```

8.3.2 Install Oracle8i Enterprise Edition Database Server

Install the Oracle8i Enterprise Edition on the remote database server, which will host the WAS and WCS databases.

Complete the following steps to install the Oracle8i Enterprise Edition server:

1. Start a Solaris Console window, and log in as user oracle.

```
# su - oracle
```

2. Insert the Oracle8i Enterprise Edition Release 2 (8.16) for Sun SPARC Solaris CD-ROM on the database server.
3. To start the Oracle8i install, type the following:

```
# xhost +
access control disabled, clients can connect as any host
# su - oracle
$ cd /cdrom/oracle8i
$ ./runInstaller
```

4. When Oracle8i Welcome window appears, click **Next**.
5. When the File Locations window appears, verify that the paths are as follows and then click **Next**:
   - Source: `/cdrom/oracle8i/stage/products.jar`
   - Destination: `/opt/oracle8/u01/app/oracle/product/8.1.6`

   Note: The destination should be the path specified as the ORACLE_HOME in, “Set environment variables” on page 227.

6. When the Inventory Location window appears, accept the default and click **OK**.

   For example, base directory: /opt/oracle8/u01/oraInventory
7. When the UNIX Group Name window appears, enter the group `oinstall` (defined above), and then click **Next**.
8. When the Oracle Universal Installer window appears, you will be prompted to run a script as user root in another Console window.
   a. Start Console window and log in as root.
   b. Execute script by typing the following:

      ```
      # /tmp/OraInstall/orainstRoot.sh
      ```

   c. You should see a message that states the following:

      ```
      Creating Oracle Inventory pointer file (/var/opt/oracle/orainst.loc).
      Changing groupname of /opt/oracle8/u01/oraInventory to dba.
      ```

9. Click the Oracle Universal Install window to bring it back to the foreground, and then click **Retry**.
10. When the Available Products window appears, select **Oracle8i Enterprise Edition 8.1.6.0.0**, and then click **Next**.
11. When the Installation Types window appears, select **Custom**, and then click **Next**.
12. When the Available Product Components window appears, do the following:

- Under Oracle8i Server 8.1.6.0.0, select and expand **Optional**, and deselect **Legato Storage Manager**
- Deselect **Oracle Product Options 8.1.6.0.0**
- Deselect **Development Tools 8.1.6.0.0**
- Select **Oracle Java Products 8.1.6.0.0 -> Oracle JDBC drivers**, select the following drivers, and then click **Next**:
  - Select **Oracle JDBC/OCI Driver for JDK 1.1 8 1.6.0.0**
  - Select **Oracle JDBC/OCI Driver for JDK 1.2 8.1.6.0.0**
  - Select **Oracle JDBC Thin Driver for JDK 1.1 8 1.6.0.0**
  - Select **Oracle JDBC Thin Driver for JDK 1.2 8.1.6.0.0**

You should see the status bar indicating the components are loading.

13. When the Component Locations window appears, accept the default and click **Next**.

    You should see the status bar indicating loading components.

14. When Privileged Operating System Groups window appears, accept the default and click **Next**. In our example, the Database Administrator and Database Operator are in group dba.

    You should see the status bar indicating the components are loading.

15. When the Create Database window appears, select **No** and then click **Next**.

> **Note:** We will create databases for WAS and WCS in subsequent steps via the Database Configuration Assistant.
>
> The installation takes approximately 30 minutes to complete.

16. When the Setup Privileges window appears, a message will be displayed to run the following script:
    a. Start a Console window and log in as root.
    b. Execute the script by typing:

    ```
    # /opt/oracle8/u01/app/oracle/product/8.1.6/root.sh
    ```

    c. You will be prompted to enter the full path to the local bin directory (we entered `/usr/bin`) and then press Enter.
    d. Click the **Setup Privileges** window to bring it to the foreground, and then click **OK**.
17. When the Net8 Configuration Assistant Welcome window appears, click **Next**.

18. When the Net8 Configuration Assistant: Directory Service Access window appears, click **No, I want to defer directory service access configuration to another type**.
19. When the Net8 Configuration Assistant: Listener Configuration Listener Name window appears, accept the default (LISTENER) and click **Next**.
20. When the Net8 Configuration Assistant: Listener Configuration Select Protocols window appears, accept the default (TCP in selected protocols) and click **Next**.
21. When the Net8 Configuration Assistant: Listener Configuration TCP/IP Protocols window appears, accept the default (use the standard port number of 1521)) and click **Next**.
22. When the Net8 Configuration Assistant: Listener Configuration Select Protocols window appears, accept the default (TCP in selected protocols) and click **Next**.
23. When the Net8 Configuration Assistant: Listener Configuration More Listeners window appears, accept the default (No additional listeners) and click **Next**.
24. When the Net8 Configuration Assistant: Listener Configuration Done window appears, click **Next**.
25. When the Net8 Configuration Assistant: Naming Methods Configuration window appears, select **No, I do not want to change the naming methods configured**, and then click **Next**.
26. When the Net8 Configuration Assistant Done window appears, click **Finish**.
27. When the End of Installation window appears, click **Exit**.

The Oracle8i Enterprise Edition server installation is now complete.

8.3.3 Oracle8i post install configuration

After the Oracle8i installation, we need to perform the following configuration steps:

- Update the oracle user .profile file
- Modify dbstart script
- Create dbora file for database autostart (optional)
- Net8 configuration

Update the oracle user .profile file

In this section we will add the Oracle8i required environment variables to the oracle user .profile file, by completing the following steps:

1. Start the Solaris Console window.
2. Log in as the oracle user.

   ```
   # su - oracle
   ```

3. Add entries to the end of the oracle user .profile file as seen in Example 8-1

Example 8-1 oracle user .profile file entries required for Oracle8i

```
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/product/jlib

ORACLE_SID=wcs
ORAENV_ASK=NO
. /usr/bin/oraenv
```

Modify dbstart script

Depending on the Oracle8i installation, the dbstart script may have problems identifying the version of Oracle8i. For this reason, we modified the dbstart script to specify the version.

1. Log in as user oracle:

   ```
   # su - oracle
   ```

2. Change directory to $ORACLE_HOME/BIN:

   ```
   $ cd $ORACLE_HOME/bin
   ```

3. Back up original dbstart script:

   ```
   # cp dbstart dbstart.orig
   ```

4. Modify dbstart script:
   a. Search for the following:

      ```
      STATUS=1
      if [ "Version" = "8.1" ]
      ```

   b. Above the lines in the previous step, add the following:

      ```
      VERSION="8.1"
      ```

   c. Save and close the file.

Create dbora file for database autostart (optional)

Refer to the *Oracle8i Installation Guide, Release 2 (8.16) for Sun SPARC*, A77181-01 for more detailed instructions on configuring the autostart of databases.

Create a file named dbora in the /etc/init.d directory. We have included our modified version of this file in Example 8-2.

1. We changed the ORA_HOME to reflect the directory specified during the Oracle8i server installation.

Example 8-2 Sample dbora file

```
#!/bin/sh
# Set ORA_HOME to be equivalent to the ORACLE_HOME
# from which you wish to execute dbstart and
# dbshut
# set ORA_OWNER to the user id of the owner of the
# Oracle database in ORA_HOME
ORA_HOME=/opt/oracle8/u01/app/oracle/product/8.1.6
ORA_OWNER=oracle

if [! -f $ORA_HOME/bin/dbstart]
then
echo "Oracle startup: cannot start"
exit
fi

case "$1" in
'start')
# Start the Oracle databases:
# The following command assumes that the oracle login will not prompt the
# user for any values

su - $ORA_OWNER -c $ORA_HOME/bin/dbstart &
su - $ORA_OWNER -c '/opt/oracle8/u01/app/oracle/product/8.1.6/bin/lsnrctl
start' &

;;
'stop')
# Stop the Oracle databases:
# The following command assumes that the oracle login will not prompt the
# user for any values
su - $ORA_OWNER -c $ORA_HOME/bin/dbshut &
;;
esac
```

2. Link the dbora file by typing the following commands:

```
# ln -s /etc/init.d/dbora /etc/rc0.d/K10dbora
# ln -s /etc/init.d/dbora /etc/rc2.d/S99dbora
```

Net8 configuration

This section describes the required configuration for Net8 after the Oracle8i installation.

1. Start the Solaris Console and log in as root.

2. Update the /etc/services file with the listener. The listener name and port where defined during the Oracle8i installation (Net8).

   We added the following line to our services file:

   ```
   LISTENER 1521/tcp
   ```

3. Save the services file.

   Note: The /etc/services file is symbolically linked to /etc/inet/services. When saving the file with vi, use the vi w! to override the default write permissions.

4. Check the status of the listener:

   ```
   # su - oracle
   $ lsnrctl status LISTENER
   ```

   Where `LISTENER` is the name or your listener.

8.3.4 Prepare Oracle8i databases for WAS and WCS

Prior to the installation of the WebSphere Application Server (WAS), and WebSphere Commerce Suite (WCS), the WAS and WCS databases need to be created.

The database preparation is organized into the following sections:

- Create an Oracle8i database
- Oracle8i database verification
- Database tuning
- Configure database autostart
- Create WAS Oracle ID and tablespace
- Create WCS Oracle ID and tablespace
- Configure Net8 on Oracle8i Server

Create an Oracle8i database

Create the following Oracle8i databases in preparation for the WAS and WCS installation. Repeat the database creation process for both databases.

- was (WebSphere Application Server repository database)
- wcs (WebSphere Commerce Suite application database)

To create an Oracle8i database, complete the following steps on the Oracle8i Enterprise Edition Server system:

1. Start the Solaris Console window and log in as user oracle.

2. Run the Oracle8i Database Configuration Assistant:

   ```
   $ dbassist &
   ```

   Note: You may need to increase the Solaris Kernel settings found in /etc/system to run dbassist. This is dependent on system memory available and current settings for processes.

3. When the Oracle Database Configuration Assistant welcome window appears, select **Create a database** and then click **Next**.
4. Select **Custom** as the database type and then click **Next**.
5. Select **Multipurpose** as the application type and then click **Next**.
6. Enter the Concurrently connected users: <value>. We accepted the default (15), and then clicked **Next**.
7. Select **Dedicated Server Mode** as the server mode and then click **Next**.
8. In the Select Options window deselect all options, except **SQL *Plus help** (optional), and then click **Next**.
9. When asked to review the database information, enter the following:
   - Global Database Name: `was`
   - SID: `was`
   - Initialization Filename: /opt/oracle8/u01/admin/was/pfile/initwas.ora
   - Compatible Parameter: 8.0.5
   - Click **Change Character Set**:
     i. Select the Character Set pull-down and select **UTF8**.
     ii. Select the National Character Set pull-down and select **UTF8**.
     iii. Click **OK**.
   - Click **Next**.
10. When the Review the Control File Parameters window appears, accept the defaults and click **Next**.
11. When the Review the System Tablespace Information window appears, accept the defaults and click **Next**.
12. When the Review the Redo Log File Parameter Information window appears, accept the defaults and click **Next**.
13. When the Review the Logging Parameter Information window appears, accept the defaults and click **Next**.
14. When the Review the SGA Information window appears, accept the defaults and click **Next** (tuning will be done in later steps).

15. When the Review the Directory Path Information window appears, accept the defaults and click **Next**.
16. Select **Create database now**, and click **Finish**.
17. When the Oracle Database Configuration Assistant alert appears, asking if you want to proceed, click **Yes**.

    The database creation progress indicator should be visible. This process takes approximately 20 minutes.
18. Repeat the database creation process for the wcs database.

Oracle8i database verification

After the installation, we recommend that you take some steps to verify that the Oracle8i Enterprise Edition server is installed correctly before attempting to configure a database client for remote connectivity.

To verify the Oracle8i Enterprise Edition, complete the following steps:

1. Start the Console and log in as user oracle.

   ```
   # su - oracle
   ```
2. To start an Oracle command window, type the following:

   ```
   $ sqlplus system/manager@was
   SQL> quit
   $ sqlplus system/manager@wcs
   SQL> quit
   ```

   Syntax:

   > sqlplus system/<system_password>@<SID>

Database tuning

1. Start a Console window and log in as user oracle.

   ```
   # su - oracle
   ```
2. Modify the following parameters in the $ORACLE_HOME/dbs/init<your_SID>.ora initialization file:

Table 8-4 Oracle8i database tuning parameters

Parameters	WAS database value	WCS database value
open_cursors	200	200
db_block_buffers	2000	3750

Parameters	WAS database value	WCS database value
shared_pool_size	20000000	30000000
processes	100	100

Note: In our example, we use one Oracle8i database for the WebSphere Application Server repository database (was), and another of the WebSphere Commerce Suite application database (wcs).

3. Stop and start the Oracle8i Server for database tuning changes to take effect.

```
# su - oracle
$ dbshut
$ dbstart
```

   Note: The Oracle8i dbshut and dbstart scripts will stop and start all databases. This behavior is desired in this example, but may not be appropriate in other situations.

   Using `svrmgrl` is another option for stopping and starting the databases.

4. If your Oracle8i Server database instances do not start, check the tuning parameters entered for errors. Changing the tuning values beyond the physical memory available on the system will result in the database instance not starting.

Configure database autostart

The following procedure will automate the startup and shutdown of the Oracle8i database server.

1. Edit the /var/opt/oracle/oratab file.

   Database entries in the oratab file appear in the following format:

   `ORACLE_SID:ORACLE_HOME:{Y|N}`

   Where `Y` or `N` specifies whether you want the dbstart and dbshut scripts to start up and shut down the database. Find the entries for all the databases that you want to start up. They are identified by the `SID` in the first field. Change the last field for each to Y to enable the database to start when the **`dbstart`** command is run.

   For example:

   `wcs:/opt/oracle8/u01/app/oracle/product/8.1.6:Y`

2. Create the dbora file.

This procedure only needs to be completed one time on the Oracle8i database server.

a. Change to the /etc/init.d directory:

```
# cd /etc/init.d
```

b. Create a file called dbora.

Example 8-3 provides a sample dbora file. We changed the ORA_HOME to reflect the directory specified during the Oracle8i server installation.

Example 8-3 Sample dbora with ORA_HOME

```
#!/bin/sh
# Set ORA_HOME to be equivalent to the ORACLE_HOME
# from which you wish to execute dbstart and
# dbshut
# set ORA_OWNER to the user id of the owner of the
# Oracle database in ORA_HOME
ORA_HOME=/opt/oracle8/u01/app/oracle/product/8.1.6
ORA_OWNER=oracle
if [! -f $ORA_HOME/bin/dbstart]
then
echo "Oracle startup: cannot start"
exit
fi
case "$1" in
'start')
# Start the Oracle databases:
# The following command assumes that the oracle login will not prompt the
# user for any values
su - $ORA_OWNER -c $ORA_HOME/bin/dbstart &
;;
'stop')
# Stop the Oracle databases:
# The following command assumes that the oracle login will not prompt the
# user for any values
su - $ORA_OWNER -c $ORA_HOME/bin/dbshut &
;;
esac
```

c. Link the dbora file as follows:

```
# ln -s /etc/init.d/dbora /etc/rc0.d/K10dbora
# ln -s /etc/init.d/dbora /etc/rc2.d/S99dbora
```

Create WAS Oracle ID and tablespace

To create a WAS Oracle user ID and tablespace, complete the following steps:

1. Start the Solaris Console window on the database server system, and log in as user oracle (Oracle DBA).

2. Start SQL Plus session by typing the following command:

```
$ sqlplus system/manager@was
```

   Syntax:

   > sqlplus system/manager@<was_SID>

3. Create the WAS tablespace from the sqlplus session, by typing the following commands:

```
SQL> CREATE TABLESPACE was
DATAFILE '%ORACLE_HOME%\DATABASE\was.ora'
SIZE 4M
REUSE
AUTOEXTEND ON NEXT 2M
MAXSIZE UNLIMITED;
```

4. Create the WAS user ID ejsadmin from the sqlplus session, by typing the following commands:

```
SQL> CREATE USER ejsadmin
IDENTIFIED BY ejsadmin
DEFAULT TABLESPACE was
QUOTA UNLIMITED ON was;
```

5. Grant privileges to the WAS Oracle user ejsadmin by typing the following command:

```
SQL > GRANT dba TO ejsadmin;
SQL> ALTER USER ejsadmin TEMPORARY TABLESPACE temp;
```

6. Create the WAS user ID ejb from the sqlplus session, by typing the following commands:

```
SQL> CREATE USER ejb
IDENTIFIED BY ejb
DEFAULT TABLESPACE was
QUOTA UNLIMITED ON was;
```

7. Grant privileges to the WAS Oracle user ejb by typing the following command:

```
SQL > GRANT dba TO ejb;
SQL> ALTER USER ejb TEMPORARY TABLESPACE temp;
```

Create WCS Oracle ID and tablespace

To create a WCS Oracle user ID and tablespace, complete the following steps:

1. Start the Solaris Console window on the database server system, and log in as user oracle (Oracle DBA).
2. Start SQL Plus session by typing the following command:

```
$ sqlplus system/manager@wcs
```

   Syntax:

```
> sqlplus system/manager@<wcs_SID>
```

3. Create the WCS tablespace from the sqlplus session, by typing the following commands:

```
SQL> CREATE TABLESPACE wcs
DATAFILE '%ORACLE_HOME%\DATABASE\wcs.ora'
SIZE 4M
REUSE
AUTOEXTEND ON NEXT 2M
MAXSIZE UNLIMITED;
```

4. Create the WCS user ID from the sqlplus session, by typing the following commands:

```
SQL> CREATE USER wcsadmin
IDENTIFIED BY wcsadmin
DEFAULT TABLESPACE wcs
QUOTA UNLIMITED ON wcs;
```

5. Grant privileges to the WCS Oracle user by typing the following command:

```
SQL > GRANT dba TO wcsadmin;
SQL> ALTER USER wcsadmin TEMPORARY TABLESPACE temp;
```

Verify user creation

To verify the users have been corrected correctly, log on to sqlplus with each of the user IDs as follows:

1. Log in as user oracle:

```
# su - oracle
```

2. Start sqlplus using ejsadmin:

```
$ sqlplus ejsadmin/ejsadmin@was
SQL> quit
```

3. Start sqlplus using ejb:

```
$ sqlplus ejb/ejb@was
SQL> quit
```

4. Start sqlplus using wcsadmin:

```
$ sqlplus wcsadmin/wcsadmin@wcs
SQL> quit
```

Configure Net8 on Oracle8i Server

To configure Net8 on the Oracle8i Enterprise Edition database server, complete the following steps:

1. Start the Solaris Console window.

2. Disable access control for XWindows display by user root by typing the following:

   ```
   # xhost +
   ```

3. Log in as user oracle:

   ```
   # su - oracle
   ```

4. Start Net8 Assistant:

   ```
   # netasst
   ```

5. Verify that Service Names exist for the following:
   - was
     - Service Name: was
     - Protocol: TCP/IP
     - Host Name: sun2
     - Port Number: 1521
   - wcs
     - Service Name: wcs
     - Protocol: TCP/IP
     - Host Name: sun2
     - Port Number: 1521

6. Check the status of the listener:

   ```
   $ lsnrctl status <listener_name>
   ```

   For example:

   ```
   $ lsnrctl status LISTENER
   ```

   Note: All listeners' statuses will be displayed if the listener name is omitted.

7. If the listener is not started, start as follows:

   ```
   $ lsnrctl start <listener_name>
   ```

   For example:

   ```
   $ lsnrctl start LISTENER
   ```

   Note: All listeners will be started if the listener name is omitted.

8.4 Install the Oracle8i Client

This section provides detailed instructions for installing the Oracle8i Client. This procedure is performed on the WCS Server system.

The Oracle8i Client installation is organized into the following phases:

- Oracle8i Client base installation
- Oracle8i Net8 client configuration
- Update oracle .profile file on Oracle8i Client
- Oracle8i Client install verification

Oracle8i Client base installation

To install Oracle8i Client, complete the following steps:

1. Complete Oracle8i prerequisites detailed in 8.3.1, “Oracle8i install prerequisites” on page 222.
2. Start a Solaris Console window, and log in as user oracle from the CDE.
   a. Click **Exit** from the Front Panel to log out as user root.
   b. Log in to the CDE as user oracle.

   Note: When logged in as user oracle using the following method, the installer may throw an exception if it is not correctly able to access the display.

   ```
   # su - oracle
   # export DISPLAY=<hostname>:0.0
   ```

   To avoid this problem, we logged into the CDE as user oracle.

3. Insert the Oracle8i Enterprise Edition Release 2 (8.16) for Sun SPARC Solaris CD-ROM in the database server CD-ROM drive.
4. Start the Oracle8i install and type the following:

   ```
   # cd /cdrom/oracle8i
   # ./runInstaller
   ```

5. When the Oracle8i Welcome window appears, click **Next**.
6. When the File Locations window appears, verify that the paths are as follows and then click **Next**.
   – Source: `/cdrom/oracle8i/stage/products.jar`
   – Destination: `/opt/oracle8/u01/app/oracle/product/8.1.6`

Note: The destination should be the path use specified as the ORACLE_HOME in “Set environment variables” on page 227.

7. When the Inventory Location window appears, accept the default (for example, base directory: /opt/oracle8/u01/oraInventory) and click **OK**.
8. When the UNIX Group Name window appears, enter the group defined above (for example enter `dba`) and then click **Next**.
9. When the Oracle Universal Installer window appears, you will be prompted to run a script as user root in another Console window.
   a. Start a Console window and log in as root.
   b. Execute the script by typing the following:

      ```
      # /tmp/OraInstall/orainstRoot.sh
      ```

   c. You should see a message that states the following:

      ```
      Creating Oracle Inventory pointer file (/var/opt/oracle/orainst.loc).
      Changing groupname of /opt/oracle8/u01/oraInventory to dba.
      ```

10. Click the Oracle Universal Install window to bring it back to the foreground, and then click **Retry**.
11. When the Available Products window appears, select **Oracle8i Client Edition 8.1.6.0.0**, and then click **Next**.
12. When the Installation Types window appears, select **Custom**, and then click **Next**.
13. When the Available Product Components window appears, do the following:
    – Select **Oracle Java Products 8.1.6.0.0 -> Oracle JDBC drivers**, and select all the available drivers:
      - Select **Oracle JDBC/OCI Driver for JDK 1.1 8 1.6.0.0**
      - Select **Oracle JDBC/OCI Driver for JDK 1.2 8.1.6.0.0**
      - Select **Oracle JDBC Thin Driver for JDK 1.1 8 1.6.0.0**
      - Select **Oracle JDBC Thin Driver for JDK 1.2 8.1.6.0.0**
    – Click **Oracle Programmer 8.1.6.0.0** and deselect **Development Tools 8.1.6.0.0** (optional).
    – Click **Next**.

    You should see the status bar indicating the components are loading.
14. When the Component Locations window appears, accept the default and click **Next**.

    You should see the status bar indicating the components are loading.

15. When the Summary window appears, review the summary information, and then click **Install**.

    The installation takes approximately 10 minutes to complete.

16. When the Setup Privileges window appears, a message will be displayed to run the following script:

    a. Start Console window and log in as root.

    b. Execute the script by typing:

    ```
    # /opt/oracle8/u01/app/oracle/product/8.1.6/root.sh
    ```

    c. You will be prompted to enter the full path to the local bin directory, which we entered `/usr/bin` and then press Enter.

    d. Click the Setup Privileges window to bring it to the foreground, and then click **OK**.

The base installation is now complete. The Oracle8i Enterprise Edition install will automatically continue to the next section.

Oracle8i Net8 client configuration

17. When the Net8 Configuration Assistant Welcome window appears, click **Next**.

18. When the Net8 Configuration Assistant: Directory Service Access window appears, click **No, I want to defer directory service access configuration to another type**.

19. When the Net8 Configuration Assistant: Naming Methods Configuration window appears, accept the default and then click **Next**.

20. When the Net8 Configuration Assistant: Net Service Name Configuration, Database Version window appears, accept the default (Oracle8i database or service) and then click **Next**.

21. When the Net8 Configuration Assistant: Net Service Name Configuration, Service Name window appears, enter the following:

    – Service Name: `<global_database_name>` (defined in the Oracle8i Enterprise Edition Database Server install)

      For example, we entered: `was`

    – Click **Next**.

22. When the Net8 Configuration Assistant: Net Service Name Configuration, Select Protocols window appears, select **TCP** and then click **Next**.

23. When the Net8 Configuration Assistant: Net Service Name Configuration, TCP/IP Protocol window appears, do the following:

    – Host Name: `<database_server_fully_qualified_hostname>`

For example, we entered: `sun2.itso.ral.ibm.com`

- Select **Use the standard port number of 1521**

> **Note:** If you specified a different port than the default during the Oracle8i server install, use the port you defined.

- Click **Next**.

24. When the Net8 Configuration Assistant: Net Service Name Configuration, Test window appears, select **Yes, perform a test**.
25. When the Net8 Configuration Assistant: Net Service Name Configuration, Connecting window appears, you will see an error message stating that the test did not succeed. A default user ID of Scott is defined.

    Click **Change Login**, and enter the following:

    - Username: `system` (default)
    - Password: `manager` (default)
    - Click **OK**.

26. The test will occur automatically after clicking OK to change the login. You should see a message stating `Connecting... Test successful.` When done click **Next**.
27. When the Net8 Configuration Assistant: Net Service Name Configuration, Net Service Name window appears, enter the following:

    - Net Service Name: `<global_database_name>.<domain>`

      For example, `was.itso.ral.ibm.com`

    - Click **Next**.

28. When the Net8 Configuration Assistant: Net Service Name Configuration, Another Net Service Name window appears, select **Yes**, and then click **Next**.
29. Repeat steps to create wcs Net Service Name.
30. Now that you have created a was and wcs Net Server Name, when the Net8 Configuration Assistant: Net Service Name Configuration, Another Net Service Name window appears, select **No**, and then click **Next**.
31. When the Net8 Configuration Assistant: Net Service Name Configuration Done window appears, click **Next**.
32. When the Net8 Configuration Assistant: Naming Methods Configuration Done window appears, click **Next**.
33. When the Net8 Configuration Assistant Done window appears, click **Finish**.
34. When the End of Installation window appears, click **Exit**.

The Oracle8i Client install is now complete.

Update oracle .profile file on Oracle8i Client

In this section we will add Oracle8i required environment variables to the oracle user .profile file that we created by completing the following steps:

1. Start the Solaris Console window.
2. Log in as the oracle user.

   ```
   # su - oracle
   ```

3. Add entries to the end of the oracle user .profile file as seen in Example 8-4

Example 8-4 oracle user .profile file entries required for Oracle8i

```
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/product/jlib
```

4. Click **Exit** to log off and log in to CDE as user oracle.

Oracle8i Client install verification

After the installation, we recommend that you take some steps to verify that the Oracle8i client is installed correctly.

To verify the Oracle8i Client, complete the following steps:

1. Start a Console and log in as user oracle.

   ```
   # su - oracle
   ```

2. To start an Oracle command window, type the following:

   ```
   $ sqlplus system/manager@was
   SQL > quit
   $ sqlplus system/manager@wcs
   SQL > quit
   ```

   Syntax:

   > sqlplus system/<system_password>@<SID>

If the previous step worked, then the listener is started and name service has been defined correctly. Proceed to the WebSphere Application Server install.

8.5 Install the WebSphere Application Server

The WebSphere Application Server V3.5, Advanced Edition installation is organized into the following phases:

- WAS V3.5 install prerequisites
- WAS V3.5 install
- Configure port 80 for the iPlanet Web Server
- WAS configuration for Oracle8i
- Configuring the WebSphere Application Server
- WAS V3.5 install verification
- WAS V3.5 Fixpack 2 installation
- WAS V3.5 E-fixes installation required by WCS V5.1

8.5.1 WAS V3.5 install prerequisites

Before installing the WebSphere Application Server verify the following prerequisites.

- Ensure your iPlanet Web Server is stopped:

  To stop HTTP Web Server, enter:

  ```
  # cd /usr/netscape/server4/https-sun1.itso.ral.ibm.com
  # ./stop
  ```

  To stop HTTPS Web Server, enter:

  ```
  # cd /usr/netscape/server4/https-SSL
  # ./stop
  ```

- Ensure the Oracle8i server and listener are started.

8.5.2 WAS V3.5 install

The IBM WebSphere Application Server V3.5, Advanced Edition (WAS) GUI installation, executed via **`./install`**, will not work for iPlanet Web Server V4.17 due to a prerequisite check failure. For this reason, we have used the command-line version of the installation program.

To install WAS on the WCS Server system, complete the following steps:

1. Start the Solaris Console and log in as user root.
2. Insert IBM WebSphere Application Server V3.5, Advanced Edition CD-ROM into the CD-ROM drive of the WCS Server system.
3. Start the WAS V3.5 install program by typing the following:

   ```
   # cd /cdrom/adv_sun_128/spool
   # ./WebSphereInstallSUN.sh
   ```

4. When prompted for the Installation Type, enter `2` for Custom Installation, and then press Enter.
5. When prompted for the directory where WebSphere will be installed, accept the default (/opt) and press Enter.

   Note: The installer will create a directory tree within this directory.

6. When prompted for the directory where WebSphere install packages are located, enter the following and then press Enter:
   - Packages Directory: `/cdrom/adv_sun_128/spool`
7. When prompted for security information, enter the following and press Enter:
   - User Name: `wasadmin`
   - Password: `<your_password>`
8. When prompted for the WebSphere install packages, enter the number and then press Enter to select the package. Repeat this process until all of the following packages are selected:
   - Application and Administrative Server
   - Administrative Console
   - Samples
   - Help
   - Configure Default Server and Applications
   - WebServer Plug-ins

   When all the desired packages have been selected, press Enter to continue.
9. When prompted to enter a Web server plug-in to install, type `2` for iPlanet 4.0 and then press Enter.
10. When prompted for the location of configuration files, enter the following and then press Enter:
    - `iPlanet Web Server config path (obj.conf):`
      `/usr/netscape/server4/https-SSL/config`

    Note: This will automatically update the HTTPS (secure) server configuration file as part of the installation.

    The configuration of the HTTP (non-secure) server will be done manually after the WAS install.

11. When prompted to specify if the Web Server is secure, enter `Y` for Secure and then press Enter.
12. When prompted for language documentation, enter `1` for English and then press Enter.
13. When prompted for the database type, enter `2` for Oracle and then press Enter.
14. When prompted for the database name, enter `was` and press Enter.

    Note: The WebSphere Application Server repository database name is user defined and can be different than the name was. Within the context of Oracle a WebSphere Application Server database is the equivalent of a tablespace with the Oracle database.

15. When prompted to enter the database home directory ($ORACLE_HOME), enter the following and then press Enter:
    - Database home directory: `/opt/oracle8/u01/app/oracle/product/8.1.6`
16. When prompted to enter the database user, enter the following and then press Enter:
    - User: `ejsadmin` (user created , “Create WAS Oracle ID and tablespace” on page 237)
    - Password: `<your_password>`
    - Confirm Password: `<your_password>`
17. When prompted to enter the Database JDBC URL, entering the following and then press Enter:
    - Database URL: jdbc:oracle:thin@<hostname>:<server_port>:<db_SID>

      For example: jdbc:oracle:thin@sun2:1521:was
18. When prompted to save information to be used for a silent installation, enter `N` for no and press Enter.
19. Press Enter to continue to begin installation of files.

    Check the log file at /tmp/WebSphere.instl
20. Tail the log file to check the installation:

    # tail -f /tmp/WebSphere.instl

Note: At the end of the WebSphere.instl log file you may see the following error message:

```
../sun/olt/dgbsetup: cannot execute.
```

Ignore this message.

The IBM WebSphere Application Server V3.5, Advanced Edition installation is now complete.

8.5.3 Configure port 80 for the iPlanet Web Server

After the installation of WAS V3.5, the iPlanet Web Server port 80 will need to be configured manually. The secure iPlanet Web Server (443) was configured during the WAS V3.5 installation.

1. Change to the directory of the HTTP Server (non-secure, port 80) obj.conf found in the /usr/netscape/server4/https-<hostname>/config directory. For example:

   ```
   # cd /usr/netscape/server4/https-sun1.itso.ral.ibm.com/config
   ```

   Note: To ensure that you add the lines at the correct location, use the secure obj.conf file (port 443) as a reference. For example:

   ```
   /usr/netscape/server4/https-SSL/config/obj.conf
   ```

2. Insert the following two lines above "<object name=default>":

   ```
   Init fn="load-modules"funcs="init_exit,auth_exit,service_exit,term_exit"
   shlib="/opt/IBMWebAS/bin/libns40.so"
   Init fn="init_exit"
   bootstrap.properties="/opt/IBMWebAS/properties/bootstrap.properties"
   ```

3. Add the following line immediately below <object name=default>:

   ```
   NameTrans from="/IBMWebAS"fn="pfx2dir"
   dir="/opt/IBMWebAS/web"
   ```

4. Add the following line immediately below PathCheck:

   ```
   fn="unix-uri-clean":
   PathCheck fn="auth_exit"
   ```

5. Add the following line immediately below ObjectType fn="force-type":

   ```
   type="text/plain":
   service fn="service_exit"
   ```

6. Remove the following line:

```
NameTrans fn="pfx2dir"from="/servlet"
dir="/usr/netscape/server4/docs/servlet"
name="ServletbyExt"
```

7. Verify the HTTP Web Server (port 80) is working properly:
   a. Start server:

```
# cd /usr/netscape/server4/https-sun1.itso.ral.ibm.com
# ./start
```

   b. Enter the following URL in a Web browser: `http://<hostname>`

8.5.4 WAS configuration for Oracle8i

Prior to starting the WebSphere Application Server, we need to complete the following configuration steps:

1. Start the Solaris Console and log in as root.
2. Change to the WebSphere Application Server directory /opt/IBMWebAS/bin:

```
# cd /opt/IBMWebAS/bin
```

3. Copy startupServer.sh to startupServer.sh.orig:

```
# cp startupServer.sh startupServer.sh.orig
```

4. Modify startupServer.sh:
   a. Search for the following:

```
DB_INSTANCE_HOME=fully_qualified_path_to_your_Oracle_home_directory
export DB_INSTANCE_HOME
```

      If the path ends in a slash (..../.), remove it from the path.

   b. Search for the following:

```
if ["${DB_TYPE}"!="DB2"]
then
{
LD_LIBRARY_PATH=$WAS_HOME/lib/odbc/lib:$WAS_HOME/bin:
$WAS_HOME/lib:$JAVA_HOME/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
${JAVA_EXE?}\
-classpath $CLASSPATH \
-DDER_DRIVER_PATH=$DER_DRIVER_PATH \
com.ibm.ejs.sm.util.process.Nanny admin.config
}
```

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

```
if ["${DB_TYPE}"!="DB2"]
```

```
then
{
ORACLE_HOME=Oracle_home_directory
export ORACLE_HOME
LD_LIBRARY_PATH=$WAS_HOME/lib/odbc/lib:$WAS_HOME/bin:
$WAS_HOME/lib:$JAVA_HOME/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
LIBPATH=$LD_LIBRARY_PATH
export LIBPATH
NLS_LANG=AMERICAN_AMERICA.UTF8
export NLS_LANG

${JAVA_EXE?}\
-classpath $CLASSPATH \
-DDER_DRIVER_PATH=$DER_DRIVER_PATH \
com.ibm.ejs.sm.util.process.Nanny admin.config
}
```

Where `Oracle_home_directory` is the fully qualified path to your Oracle home directory. For example:

/opt/oracle8/u01/app/oracle/product/8.1.6 .

d. Save the changes.

8.5.5 Configuring the WebSphere Application Server

This section includes some general configuration steps for the WebSphere Application Server.

Start the WebSphere Application Server

To start the WebSphere Application Server, complete the following steps:

1. Starta Console window and log in as user root.
2. Start WebSphere Application Server:

```
# cd /opt/IBMWebAS/bin
# ./startupServer.sh
```

3. View the tracefile log as follows:

```
# cd /opt/IBMWebAS/logs
# tail -f tracefile
```

Look for the following message if successfully started:

```
AdminServer   A Initializing WebSphere Administration server
AdminServer   A WebSphere Administration server open for e-business
```

Start the WebSphere Administrator's Console

1. Start Console window and log in as user root.
2. Start the WebSphere Administrator's Console:.

```
# cd /opt/IBMWebAS/bin
# ./adminclient
```

3. The WebSphere Administrato'rs Console should be displayed.

#Add host aliases

Verify and add the following host aliases to WAS V3.5 via the Administrator's Console.

1. From the Administrator's Console, select **default_host**.
2. Click the **Advanced** tab.
3. In the Aliases field, ensure the aliases listed in Table 8-5 exist.

Table 8-5 WAS alias samples

Alias syntax	Alias example
<host.domain.com>	sun1.itso.ral.ibm.com
<host>	sun1
<ip_address>	9.24.105.46
<host.domain.com>:443	sun1.itso.ral.ibm.com:443
<host>:443	sun1:443
<ip_address>:443	9.24.105.46:443

Once the host aliases have been added, stop and start the application server for the changes to take effect.

8.5.6 WAS V3.5 install verification

After the installation of WAS V3.5 it is very important that you verify that the server is working properly before continuing to the next section.

1. Ensure that the admin.config, WebSphere Application Server configuration file, has the following entries:
   a. Change to the directory of the admin.config file

```
# cd /opt/IBMWebAS/bin
```

   b. Verify the JDBC URL:

```
com.ibm.ejs.sm.adminServer.dbUrl=jdbc:oracle:thin:@host_name :1521:SID
```

For example:

```
com.ibm.ejs.sm.adminServer.dbUrl=jdbc:oracle:thin:@sun2.itso.ral.ibm.com:
:1521:wcs
```

c. Verify that the classpath com.ibm.ejs.sm.adminServer.classpath= contains the path $ORACLE_HOME/jdbc/lib/classes12.zip .

Where $ORACLE_HOME is the fully qualified path to your Oracle home directory on your Oracle server, for example /opt/oracle8/u01/app/oracle/product/8.1.6.

2. Test the iPlanet Web Server (80):

a. Start HTTP Web Server:

```
# cd /usr/netscape/server4/https-sun1.itso.ral.ibm.com
# ./start
```

b. Enter the following URL in a Web browser: `http://<hostname>` to test the HTTP Server. For example, `http://sun1.itso.ral.ibm.com`

3. Test the secure iPlanet Web Server (443):

a. Start HTTP Web Server:

```
# cd /usr/netscape/server4/https-SSL
# ./start
```

b. Enter the following URL in a Web browser: `https://<hostname>` to test the HTTP Server, for example `https://sun1.itso.ral.ibm.com`.

4. Ensure the WebSphere Application Server is started. If it is not, type the following to start the server login as root:

```
# cd /opt/IBMWebAS/bin
# ./startupServer.sh
```

5. While the WebSphere Application Server is starting, verify the tracefile by typing the following command:

```
# cd /opt/IBMWebAS/logs
# tail -f tracefile
```

Look for the following message if successfully started:

```
AdminServer   A Initializing WebSphere Administration server
AdminServer   A WebSphere Administration server open for e-business
```

6. Start the WebSphere Administrator's Console by typing the following commands:

```
# cd /opt/IBMWebAS/bin
# adminclient.sh
```

7. Start the Default Server (application server) from within the Administrator's Console.

8. From a Web browser enter the following URLs:

```
http://<hostname>/servlet/snoop
https://<hostname>/servlet/snoop
```

9. If the snoop servlet worked properly, stop the Default Server (application server) to conserve system memory and continue to the next step.

8.5.7 WAS V3.5 Fixpack 2 installation

WebSphere Commerce Suite V5.1 requires that WebSphere Application Server V3.5 Fixpack 2 is installed.

To install the WebSphere Application Server Fixpack 2, ensure that you have unzip on your system, and then complete the following steps:

1. Start the Solaris Console window and log in as root.
2. Insert the WebSphere Commerce Suite V5.1 Pro for Solaris CD into the CD-ROM drive on the WCS Server system.
3. Copy the WAS V3.5 Fixpack 2 to your machine by typing:

```
# mkdir -p /opt/wasfp2
# cd /cdrom/wcs51/wasfix/wasfp2
# cp -r * /opt/wasfp2
```

4. Start WAS V3.5 Fixpack 2 install by typing:

```
# cd /opt/wasfp2
# ./install.sh
```

5. When prompted, enter the WebSphere Application Server directory, enter `/opt/IBMWebAS` (WAS default) and press Enter.
6. When asked if you want to install the IBM HTTP Server Fixpack, type n (we using iPlanet Web Server) and press Enter.
7. You will be prompted for your iPlanet Web Server document root. For iPlanet Web Server the default path is `/usr/netscape/server4/docs`. Then press Enter.
8. The WebSphere Application Server V3.5 Fixpack 2 installation is now complete.

8.5.8 WAS V3.5 E-fixes installation required by WCS V5.1

WebSphere Commerce Suite V5.1 requires that WebSphere Application Server V3.5 E-fixes be installed.

You can install the WebSphere Application Server E-fixes manually or using the script that is available on the IBM Web site that will automate the WebSphere Application Server E-fixes installation. We recommend that you use the script, since the manual installation method is tedious and will fail if a single error is made.

To use this script rather than the manual installation method, download the wcswasefix_unix.tar file from the following Web site and follow the directions provided in the accompanying wcswasefix_unix.pdf file:

```
http://www.ibm.com/software/webservers/commerce/wcs_pro/downloadsnc-solaris
.html
```

To install the WAS V3.5.2 E-fixes using the downloaded script, complete the following steps:

1. Insert WebSphere Commerce Suite V5.1 Pro for Solaris CD into CD-ROM on the WCS Server system. This CD includes the WAS eFixes JAR files.
2. Create a download directory as follows:

   ```
   # mkdir -p /opt/wasefix
   ```

3. Download the script from the following URL:

   ```
   http://www.ibm.com/software/webservers/commerce/wcs_pro/downloadsnc-solaris
   .html
   ```

4. Uncompress the tar file:

   ```
   # cd /opt/wasefix
   # tar -xvf wcswasefix_unix.tar
   ```

5. Change the permissions for the script to make it executable:

   ```
   # chmod 777 wcswasefix_unix.sh
   ```

6. Run the WAS E-fixes install:

   ```
   # ./wcswasefix_unix.sh
   ```

7. When prompted for the source point of the E-fixes, enter `/cdrom/wcs51/` and then press Enter.

Note: This will copy the E-fixes JAR files to the /opt/IBMWebAS/lib. In addition, the /opt/IBMWebAS/bin/admin.config file will be updated to include classpath entries for each JAR file.

The WebSphere Application Server V3.5.2, E-fixes installation is now complete.

Verify E-fixes installation

1. Verify the following JAR files have been copied to the /opt/IBMWebAS/lib directory:
   - pq42952.jar
   - pq43040.jar
   - 888452_352.jar
   - 85699_352.jar
   - 88424.jar
   - sas-r3502-1115.jar
   - 88299.jar
2. Verify the admin.config classpath has been updated to include the JAR files listed in the previous step.
3. Copy the xmlconfig.dtd to the /opt/IBMWebAS/bin directory as follows:

```
# cd /cdrom/wasfix/wasefix/WAS_3.5.2_efixes
# cp xmlconfig.dtd /opt/IBMWebAS/bin
```

Note: To install the E-fixes manually, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*.

8.6 Install the WebSphere Commerce Suite

This section provides instructions for installing WebSphere Commerce Suite V5.1, Pro Edition for Solaris.

8.6.1 WCS V5.1 install prerequisites

Ensure the following prerequisites are installed:

- iPlanet Web Server, Oracle8i, WAS, JDK 1.2.2 (installed by WAS install)
- WAS V3.5 Fixpack2
- WAS V3.5 E-fixes

8.6.2 WCS V5.1 installation

To install WebSphere Commerce Suite V5.1, complete the following steps:

1. Start the Solaris Console window and log in as user root.
2. Insert the WebSphere Commerce Suite V5.1, Pro Edition for Solaris CD into the CD-ROM drive on the WCS server system.

3. Start the Solaris Admintool, to install WCS, by typing the following command:

   ```
   # cd /bin/admintool &
   ```

4. When the Admintool window appears, from the menu bar click **Browse** -> **Software**.
5. When the Admintool Software window appears, from the menu bar click **Edit** -> **Add**.
6. When the Set Source Media window appears:
   - If you have Volume Manager installed:
     i. From the Software Location pull-down, select **CD with Volume Management**.
     ii. Enter the CD Path: `/cdrom/wcs51/WebSphereCommerceSuite`

     Where `cdrom` is the mount point for your CD-ROM drive.
     iii. Click **OK**.
   - If you do not have Volume Manager installed:
     i. From the Software Location pull-down, select **CD without Volume Management**.
     ii. Enter the Mount Point: `/cdrom_dir/wcs51/WebSphereCommerceSuite`

     Where `cdrom_dir` is the mount point for your CD-ROM drive.
     iii. Click **OK**.
7. When the Admintool: Add Software window appears, select the following packages from the left-hand pane, and then click **Add**:
   - Select **IBM WebSphere Commerce Suite Base**
   - Select **IBM WebSphere Commerce Suite – Blaze Software**
   - Select **IBM WebSphere Commerce Suite Documents**

   Note: WebSphere Commerce Suite V5.1 is automatically installed to the /opt/WebSphere/CommerceSuite directory (only supported directory).

8. During the installation, you will be prompted if you want to proceed with the installation of the package WCS base. Type y and press Enter.
9. During installation you may be prompted with the message, `Do you want to install these conflicting files [y,n,?,q]:` type `y` and press Enter.

   This will occur for each of the packages selected due to file permissions of the /opt/WebSphere directory.

10. When all packages are installed, press Enter and close all Admintool windows.

The WebSphere Commerce Suite V5.1 for Solaris installation is now complete.

8.7 Create a WCS instance

After the installation of WebSphere Commerce Suite V5.1, Pro Edition for Solaris and the prerequisites components, a WCS instance needs to be created as the environment to deploy and host a WCS store.

There are two methods for creating a WCS instance:

- Configuration Manager - WCS instance creation

  In this chapter we describe how to create the WCS instance using the Configuration Manager in a Solaris environment.

- Command line - WCS instance creation

  For details on creating the WCS instance via the command line refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*. In addition, you can reference the working example in Chapter 7, “WCS runtime for AIX: DB2 and IHS” on page 137.

8.7.1 WCS instance creation prerequisites

Prior to creating the WCS instance, ensure the following prerequisites steps have been completed:

1. Take note of your $ORACLE_HOME, and WCS database SID.

   For example our values are as follows:

   ```
   $ORACLE_HOME= /opt/oracle8/u01/app/oracle/product/8.1.6
   WCS database SID=wcs
   ```

2. Verify that the Oracle8i server, and listener are started.
3. Verify the iPlanet Web Server (HTTP, HTTPS) are started.
4. Verify the WebSphere Application Server is started.
5. Modify the WCS environment script setenv.sh for Oracle8i by completing the following steps:

   a. Start the Solaris Console and log in as root.

   b. Change to the directory of setenv.sh:

      ```
      # cd /opt/WebSphere/CommerceSuite/bin
      ```

c. Modify setenv.sh as follows:

Ensure the following line appears in the file:

```
export ORACLE_HOME=/opt/oracle8/u01/app/oracle/product/8.1.6
```

Ensure the following lines exist at the end of the file:

```
export NLS_LANG=AMERICAN_AMERICA.UTF8
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes12.zip:$CLASSPATH
CLASSPATH=$ORACLE_HOME/jdbc/lib/nls_charset12.zip:$CLASSPATH
export CLASSPATH
```

d. Save and exit.

6. Start Configuration Manager Server.

Before starting the Configuration Manager, the Configuration Manager Server must be started by completing the following steps:

a. Start a Solaris Console window and log in as root.

b. Start the Configuration Manager by typing the following:

```
# cd /opt/WebSphere/CommerceSuite/bin
# ./config_server.sh
```

Note: Do not close the terminal window where you entered the **config_server.sh** command or the Configuration Manager server will stop.

Do not run the **config_server.sh** command as a background process, because leaving the Configuration Manager server running could potentially cause a security problem.

8.7.2 Creating a WCS instance using the Configuration Manager

To create a WCS instance, do the following from the WebSphere Commerce Suite Configuration Manager:

1. Ensure the Configuration Manager Server is started.
2. Start a Solaris Console window.
3. Start the Configuration Manager as follows:

```
# cd /opt/WebSphere/CommerceSuite/bin
# ./config_client.sh &
```

4. When the Configuration Manager login window appears, enter the following and click **OK**:
   - User ID: `webadmin` (default)
   - Password: `webibm` (default)
5. When the Password message window appears, click **OK**.

Note: You will be asked to change your password the first time you log in.

6. When the Modify Password window appears, enter the following and then click **OK**:
   - Current Password: `webibm`
   - New Password: `<your_password>`
   - Confirm New Password: `<your_password>`
7. The Password Message window should appear with the message `Successfully modified password`. Click **OK**.

Note: In the remaining steps, we provide examples for our runtime environment settings. Refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris* for all the possible configuration options and values.

8. The Configuration Manager should now be displayed. Select and expand your **<hostname>**.
9. Right-click **Instances** -> **Create Instance**.
10. You are asked if you want to base this instance on an existing instance. Select **No**.
11. Select the **Instance** tab (default starting point) and enter the following:
    - Instance Name: `wcs`
    - Instance root: `/opt/WebSphere/CommerceSuite/instances/<instance_name>`

      Note: You must manually change the <instance_name> in the instance root path above.

    - Deselect the **Use Default Merchant Key checkbox** next to the Merchant Key.
    - Merchant Key: `<your_merchant_key>`

      The merchant key requires a 16-character hexadecimal value.

Note: To avoid a security risk, you must deselect use default merchant key and enter your own defined value.

    - Click **Next**.
12. On the Database tab, enter the following:

- Database administrator name: `wcsadmin`
- Database administrator password: `<wcsadmin_password>`
- Database name: `wcs`
- Database type: select **Oracle** from the pull-down
- Oracle instance userid: `oracle`
- Database user name: `wcsadmin`
- Database user password: `<wcsadmin_password>`
- Check **Set as active database**
- Check **Use remote database**
- Database server hostname: `sun2.itso.ral.ibm.com`
- Database server port: `1521`
- Click **Next**.

13. On the Languages tab, accept the default (wcs.bootstrap_multi_en.xml), and click **Next**.
14. On the Web Server tab, enter the following:
    - Hostname: `sun1.itso.ral.ibm.com`

      Enter the fully qualified hostname `<host>.<domain>.com`
    - Web Serve Type: select **Netscape iPlanet** from pull-down
    - Primary Document Root: `/usr/netscape/server4/docs`
    - Server Port: `80`
    - Authentication Mode: check **Basic**
    - Secure Server Configuration Path: `/usr/netscape/server4/https-SSL/config`
    - Non-secure Server Configuration Path: `/usr/netscape/server4/https-sun1.itso.ral.ibm.com/config`
    - Click **Next**.
15. On the WebSphere tab, enter the following:
    - Data Source Name: `WebSphere Commerce Suite Oracle DataSource wcs`
    - Port Number: `900`
    - JDBC Driver Location: `/opt/oracle8/u01/app/oracle/product/8.1.6/jdbc/lib/classes12.zip`
    - Check **Stores Web Application**
    - Check **Tools Web Application**

– Click **Next**.

16. On the Payment Manager tab, accept the defaults, and click **Next**.

    In this example, we do not configure Payment Manager.

17. On the Log System tab, accept the defaults, and click **Next**.
18. On the Messaging tab, accept the defaults, and click **Finish**.
19. When the Population Confirmation window appears, click **Yes**.

    The instance creation process takes approximately 20 minutes.

20. When the Configuration Manager is done creating the instance, you should see a message `Instance was successfully created`. Click **OK**.
21. The WCS instance creation is now complete.
22. Close Configuration Manager.
23. Stop the Configuration Manager Server to avoid security risk and reduce memory consumption.

8.7.3 Post WCS instance creation configuration

After creating the WCS instance, complete the following steps to configure the instance.

Configure iPlanet Web Server obj.conf

When using iPlanet Web Server, we need to apply changes to the obj.conf for the non-secure and secure Web Server by completing the following steps:

1. Start the iPlanet Web Server Administrator by entering the following URL in a Web browser:

   `http://sun1.itso.ral.ibm.com:8888/https-admserv/bin/index`

   Where `sun1.itso.ral.ibm.com` is the hostname of the Web Server.

2. Select the non-secure server from the list of available servers and click **Manage**.
3. Click **Apply** in the upper right-hand corner of the window.
4. Click **Load Configuration Files**.
5. A message should be displayed that you were successful. Click **OK**.
6. Select the secure server from the list of available servers and click **Manage**.
7. Click **Apply** in the upper right-hand corner of the window.
8. Click **Load Configuration Files**.
9. A message should be displayed that you were successful. Click **OK**.

Compile WCS administrative tools JSPs (optional)

Compiling the JavaServer Pages will significantly reduce the amount of time needed to load the WebSphere Commerce Suite tools until they are compiled by using them the first time. This is an option step.

To batch-compile JavaServer Pages (JSP) files, do the following:

1. Stop the iPlanet Web Server (HTTP, HTTPS) and WebSphere Application Server.

   This step is done as a precaution that users accessing the admin tools will force a compilation of a JSP by WAS during this manual compile procedure.
2. Start the WebSphere Administrative Console.
3. Select and expand **WebSphere Administrative Console -> <hostname>**
4. Rename the nodes as seen in Table 8-6. This step is necessary to remove spaces in the default naming for UNIX.

Table 8-6 WCS nodes to rename

Default	Change to
WebSphere Commerce Server - <instance_name>	wcs
WCS Stores	stores
WCS Tools	tools

Note: Nodes can be found by expanding the following in the WebSphere Administrative Console: **<host_name>** -> **WebSphere Commerce Server** - **<wcs_instance_name>** -> **WCS Web Container**.

To rename these nodes, select the node, select the **General** tab for that node, and rename the node in the Web Application Name field.

5. Stop and restart the WebSphere Commerce Server - <wcs_instance_name>
6. To mass-compile the JSP files run the following commands:

   ```
   # cd /opt/WebSphere/CommerceSuite/bin
   ```

   Pre-compile tools:

   ```
   # ./WCSJspBatchCompiler.sh -adminNodeName <adminNodeName> -serverName wcs
   -application tools
   ```

   Pre-compile WCS stores:

   ```
   # ./WCSJspBatchCompiler.sh -adminNodeName <adminNodeName> -serverName wcs
   -application stores
   ```

Where <adminNodeName> is the name of the node; usually this is the short host name of the machine. These commands are presented on separate lines for readability only; ensure that you type each one on a single line.

Note: Several errors will be logged when you perform these compiles. Ignore them.

Stop and start the iPlanet Web Server

Stop and start the iPlanet Web Server for the changes to take effect.

1. Type the following commands to stop the iPlanet HTTP Server:

```
# cd /usr/netscape/server4/https-sun1.itso.ral.ibm.com
# ./stop
# ./start
```

2. Type the following commands to stop the iPlanet HTTPS Server:

```
# cd /usr/netscape/server4/https-SSL
# ./stop
# ./start
```

Stop and start WebSphere Commerce Server

To stop and start the WebSphere Application Server for the changes to take effect, complete the following steps:

1. Start WebSphere Administrator's Console.

```
# cd /opt/IBMWebAS/bin
# ./adminclient.sh
```

2. Select and expand **WebSphere Administrative Domain** -> **sun1**.

   Where sun1 is the hostname of this server.

3. Select **WebSphere Commerce Server - wcs**

   Where wcs is the instance name.

4. Right-click and select **Stop**.

   The current status should be Stopped.

5. Right-click and select **Start**.

   If successful, the current status should be Running.

Verify the WCS instance

To verify the WCS instance is created properly, complete the following steps:

- Verify the existence of the instance XML configuration file <instance_name>.xml in the following directory:

/opt/WebSphere/CommerceSuite/instances/<instance_name>/xml

For example:

/opt/WebSphere/CommerceSuite/instances/wcs/xml/wcs.xml

- Review the contents of the instance creation log createdb.log in the following directory:

  /opt/WebSphere/CommerceSuite/instances/<instance_name>/logs

  For example:

  /opt/WebSphere/CommerceSuite/instances/wcs/logs/createdb.log

- Review the contents of the wcs.log regarding the deployment of WCS within the WebSphere Application Server database repository, found in the following directory:

  /opt/WebSphere/CommerceSuite/instances/<instance_name>/logs

  For example:

  /opt/WebSphere/CommerceSuite/instances/wcs/logs/wcs.log

  Use this log to ensure that the server has started correctly.

- Verify the WebSphere Commerce Suite Admin Console starts by entering the following URL in a Web browser: `http://<hostname>/adminconsole`

8.8 Deploy the sample store

To further verify the WCS runtime for Solaris, we deployed the InFashion sample store using Store Services (this step is optional).

Publish store from Store Services

To deploy the sample store using Store Services, complete the following steps:

1. Start Internet Explorer V5.5 on a Windows system (browser type and level required by WCS).
2. To start Store Services, enter the following URL on Internet Explorer V5.5 Web browser: `http://<hostname>/storeservices`
3. A Security Alert window may appear. Accept the certificate and click **Yes** to continue.
4. When the Store Services Logon window appears, enter the following:
   - User name: `wcsadmin`
   - Password: `wcsadmin` (default)
   - Click **Log On**

5. When the Create Store Archive window appears, enter the following:
   - In the sample pull-down select **infashion_en_US_es_ES.sar**.

     This sample includes multicultural support for English for the US, and Spanish for Spain.
   - Store archive (required): InFashion

     This is the name of the SAR file that will be generated to publish to your runtime environment. For example, InFashion.sar will be created.
   - Store directory (required): InFashion

     This is the directory that will serve as the root for this store.
   - Store owner: Default Organization (default)
   - Click **OK**
6. Check the **InFashion.sar** Store Archive in the list. Click **Publish**.
7. When the Publish Store Archive window appears, take note of the information on this window, and click **OK**.
8. You will return to the Store Services archive list window. Notice the Publish Status column of the table. It will first say Awaiting Publishing. Click **Refresh**. The publishing status will change to Publishing. If successful, the status will change to Publish completed successfully. After 10-15 minutes, you may want to click Refresh.
9. When the status changes to Publish Completed Successfully, check **InFashion.sar**, and then click **Publish Summary**.
10. When the Publish Summary window appears, click **Launch Store** to launch the sample store.
11. You will be prompted to enter the Web application Web path. Accept the default (/webapp/wcs/stores), and click **OK**.

    The first time the store loads, it will take a while due to the compilation of store JSPs.
12. Take note of the sample store URL:

```
https://sun1/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&la
ngId=-1&catalogId=10001
```

Test published store

To test the basic functionality of the store, to verify that it published correctly, and to verify the runtime environment, complete the following steps:

1. From a Web browser, (Netscape Communicator V4.7x or Microsoft Internet Explorer V5.5) enter the sample store URL. Change the URL from HTTPS to HTTP.

For example:

```
http://sun1/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001&lan
gId=-1&catalogId=10001
```

2. Verify Groups (categories) and Products.

8.9 Where to find more information

We have included many tips on Solaris and Oracle8i as well as documentation and useful URLs in the following appendixes:

- Appendix B, “Solaris tips” on page 585
- Appendix C, “Oracle tips” on page 595

Part 3

Developing a store

© Copyright IBM Corp. 2001

9

Store archive overview

In this chapter we provide a basic overview of the store archive (SAR) file. This chapter is organized into the following sections:

- Store archive overview
- Contents of a store archive
- Physical structure of a SAR file
- Creating a SAR file
- Understanding a Commerce Studio project

© Copyright IBM Corp. 2001

9.1 Store archive overview

A store archive file, SAR file, (file extension .sar) is a compressed archive file that contains all the assets necessary to create a store. It is used as a vehicle for packaging and delivering stores in a format that may be easily transferred, copied and deployed.

Store archives can be used as a base upon which to create new stores. They provide a template of basic functionality that can be extended as needed.

The SAR file also provides a source for easy deployment of the store. From a single source, all of the customizations and data for a store can be deployed to a server in a single step using Store Services. After publishing from Store Services, the store is fully functional.

In the context of development, a store can be packaged into a SAR upon completion. This archive can then be used a deployment source from which the store can be published to the staging or production servers. It also serves as a baseline if the running store is damaged.

The store archive file can be opened or created by almost any archiving utility capable of handling zip-files (.zip), such as WinZip or PowerArchiver.

9.2 Contents of a store archive

The store archive file contains all of the data needed to publish a working commerce site. The basic parts of the archive are file assets, database assets, payment assets, and a descriptor.

9.2.1 File assets

The file assets of an archive file are broken into two areas: Web assets and the property resource files. Both of these file assets are stored in the archive in ZIP files. During deployment, they are copied in mass to the Web document root.

Web assets

The Web assets are the files used to create your store pages, such as HTML, JSP files, images, graphics and include files. Web assets are grouped together as a compressed archive file in the store archive. When a store archive is published, the Web assets are published to the Web folder of the store. For example, if your store was named ITSODEMO1, these files would be copied to the directory <wcs_install_path>/stores/web/ITSODEMO1/web when the SAR file was published.

Property resource bundles

These files contain all of the text on your store pages. Property resource bundles are grouped together as a compressed archive (ZIP) file in the store archive. If your store supports more than one language or cultural locale, the ZIP file will contain bundles specific to language or cultural locale. Each language will occupy a directory of its own. When a store archive is published, the text assets are published to the appropriate directory. For example, if your store was named ITSODEMO1, these files would be copied to the directory <wcs_install_path>/stores/web/ITSODEMO1/properties when the SAR file was published.

9.2.2 Store database assets

These are XML files containing the data to be loaded into the database. This data includes campaigns, catalog and product information, command registration, currency information, fulfillment information, offering data, shipping rules, general store information, tax rules, and quantity units.

For a complete list of required data, please refer to *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*.

9.2.3 Descriptor

The descriptor is an XML file named sarinfo.xml that describes the store archive.

Note: Every store archive file must contain descriptor file.

The descriptor is the only mandatory file in the store archive. It contains the name of the compressed archive file that stores the Web assets and the name of the compressed archive file that stores the resource bundles. It also contains the names of include files and consistency checking files, as well as information about the archive file that is needed during the publishing process.

Also within the descriptor is a description of the store database asset XML files. For each XML file, the descriptor indicates the location of the file and its priority in the loading sequence.

Table 9-1 identifies the database assets which are commonly in a store archive, and indicates which of the database assets is mandatory.

Table 9-1 Store Database assets

Mandatory Database Assets	Optional Database Assets
fulfillment assets	member groups
catalog assets	store-catalog-tax assets
store assets	store-catalog-shipping assets
tax assets	quantity unit assets
tax fulfillment assets	campaign assets
shipping assets	
shipping fulfillment assets	
store default assets	
store-catalog assets	
store-fulfillment assets	
offering assets	
command assets	
currency assets	
payment assets	

Figure 9-1 shows how the contents of a store archive fit together to form the completed SAR file.

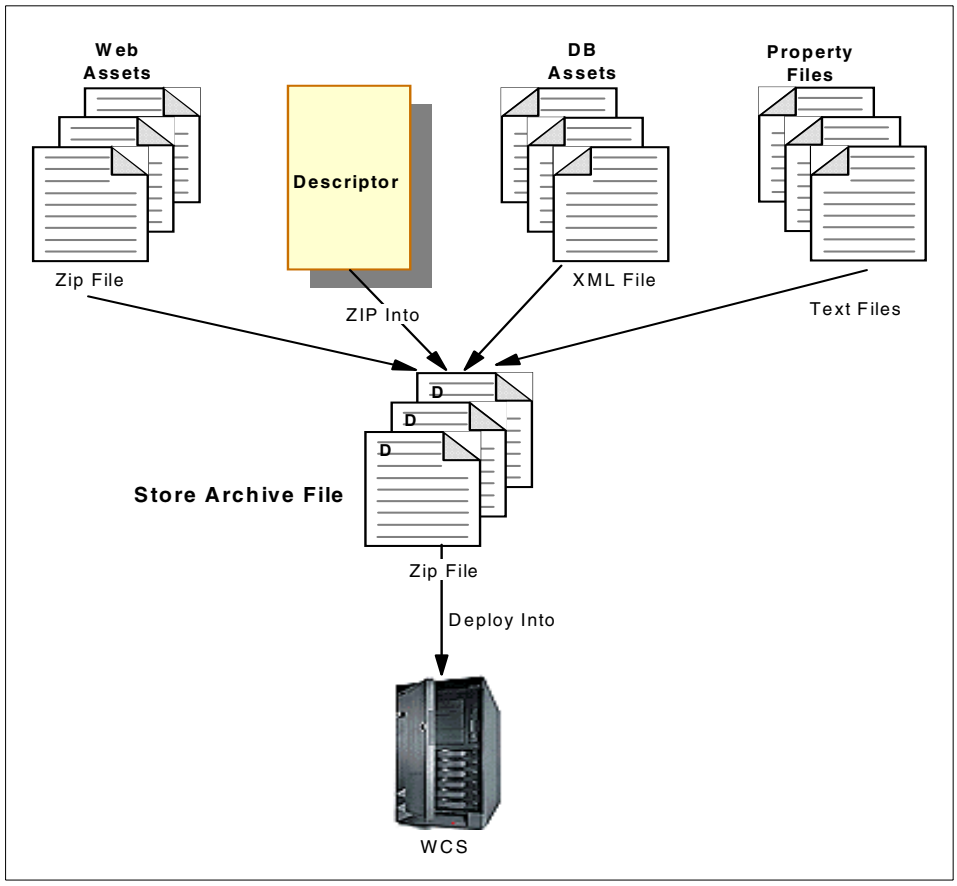


Figure 9-1 Graphical depiction of a complete SAR file

9.3 Physical structure of a SAR file

This section is divided into two subsections and outlines the recommended structure of a SAR for standard, non-multicultural store as well as for a multicultural enabled store.

Tip: We recommend opening the store archive file using an archive utility (such as WinZip) to see the contents to gain an understanding of the structure of a SAR file.

9.3.1 Standard non-multicultural store

The recommended format of a SAR file for a standard non-multicultural store is shown in Table 9-2.

Table 9-2 Structure of non-multicultural SAR file

Content type	File	Description
Descriptor	SAR-INF\sarinfo.xml	Describes the store archive. The path must be uppercase.
Web assets	webapp.zip	Contains all the Web assets for the store, including files such as JSPs, images, etc.
Properties	properties.zip	Contains the text files for the store.

Database assets	data\<name>.xml
	For instance:
	`data\catalog.xml`
	`data\store.xml`
	`data\fulfillment.xml`
		Contain all the store database assets.

Document type definition (dtd)	<name>.dtd	Contain the document type definition information for custom XML files.

Note: The descriptor (sarinfo.xml) occupies a directory of its own, and all of the database assets are extracted to the data subdirectory.

9.3.2 Multicultural store

Multicultural store archives are an extension on the structure of a non-multicultural store. The difference lies in two areas, database assets and properties.

In order to accommodate locale-specific data, an additional layer is added to the directory structure. Database assets that are specific to locale are stored in a locale-specific subdirectory under the data directory.

The text of the store also depends on cultural locale, but in this case the extension to the base SAR structure is simple. An additional resource bundle file is added to the resource bundle archive to contain data that is specific to a given cultural locale.

The recommended format of a SAR file for a multicultural-enabled store is shown in Table 9-3.

Table 9-3 Structure of a multicultural SAR file

Content type	File	Description
Descriptor	SAR-INF\sarinfo.xml	Describes the store archive.
Web assets	webapp.zip	Contains all the Web assets for the store, including files such as JSPs, images, etc.
Properties	properties.zip	Contains the text files for the store. Culture-specific files are included for each locale that a store supports.

Database assets	data\<name>.xml OR
	data\<locale>\<name>.xml
	For instance:
	`data\fulfillment.xml`
	OR
	`data\en_US\store.xml`
		Contain all the store database assets

Document type definition (dtd)	<name>.dtd	Contain the document type definition information for custom XML files.

Note: For more information on creating multicultural stores, please refer to Chapter 13, “Multicultural enablement” on page 375.

9.4 Creating a SAR file

You can create a store archive in one of the following two ways: Using Store Services or by compiling and archiving the contents yourself.

9.4.1 Using Store Services

Store Services allows you to quickly create a store archive based on the sample provided with Commerce Suite. The sample store archive creates a store based on the shopping flow most commonly used in today’s top sites, and includes the necessary JavaServer Pages and commands for a working

store. Once you have created your store archive, you can choose to publish it to the WebSphere Commerce Server immediately, thus creating a running store, then customize it, using the Store Profile, Tax and Shipping notebooks, and WebSphere Commerce Studio. Or if you prefer, you can customize the store archive, using these tools, before publishing.

We will provide a sample for this approach.

Note: Store profile, tax and shipping notebooks within Store Services can only change existing information in a store archive, not create it.

9.4.2 Building your own SAR

If the sample store archive does not meet your needs, or if you need to heavily customize the sample store to use as a base for creating other stores, you may prefer to build your own store archive. Building your own store archive requires advanced knowledge of the Commerce Suite database, XML, Java, and JavaServer Pages technology. The specifications store archives must adhere to are outlined in the WCS online help in Build a store archive.

We demonstrate how to build your own SAR in Chapter 12, “Creating a store archive (SAR) for deployment” on page 355.

9.4.3 Publishing a SAR file

There are three approaches for publishing a store:

- Publish via Store Services

  For an example, refer to 6.7.1, “Publish store from Store Services” on page 115.

- Publish via command line

  For an example, refer to 7.7.2, “Deploy the sample store via the command line” on page 175.

- Publish via WebSphere Studio

  For an example, refer to 11.5, “Publishing from Studio” on page 346.

9.5 Understanding a Commerce Studio project

Once the development environment is set up (see details in Chapter 10, “Development environment” on page 283), it is important to understand the concepts related to creating a project with the development tools installed and configured in previous steps.

In this section we address the following topics:

- Commerce Studio file structure
- Data flow between SAR and WebSphere Commerce Studio

9.5.1 Commerce Studio file structure

When a Commerce Studio store archive project is created, it follows a specific file structure in WebSphere Studio. The project must maintain this file structure in order to be exported properly back to the store archive on the WebSphere Commerce Server. So if you add any new files to the project, they *must* follow this structure.

All Web asset files in the store archive are contained in the project for that store. The project is broken down into the following folders:

- Resources - WebSphere Studio automatically creates a resources folder for all projects.
- Rules - WebSphere Studio automatically creates a rules folder for all projects.
- Servlet - WebSphere Studio automatically creates a servlet folder for all projects. If you create servlets for use in your store, store them in this folder. In all other cases, the servlet folder will be empty.
- Stores - Contains the store archive project file, named store.sapf, which is produced when you create a Studio project using the Store Archive template, and the store directory folder, where store directory is defined using the Create Store Archive page in Store Services. The store directory folder contains all HTML and JavaServer Pages files, and the following sub folders:
  - Locale - For example, en_US or de_DE. The locale folder(s) contain locale-specific images. If your store supports multiple locales, there will be multiple locale folders.
  - Images - Contains image files used in the store. You can add new images to this folder, or create new folders of your choice, under the stores folder.
- Theme - Contains the Master.css file. WebSphere Studio automatically creates a theme folder and Master.css file for all projects.

Note: If you wish to create your own store assets and publish them using WebSphere Studio, you must save your files in the stores folder according to the above file structure in order to publish properly.

9.5.2 Data flow between SAR and WebSphere Commerce Studio

We provide an overview of the data flow between a SAR file and WebSphere Commerce Studio to demonstrate what files are imported to what folder (see Table 9-5).

Table 9-4 Data flow from SAR to WebSphere Commerce Studio

Content type and description	Target folder in WC Studio	Comment
SAR-INF sarinfo.xml	None by default - you have to import it manually	Create a folder in Studio
Webapp webapp.zip	stores folder	No additional action needed
Properties properties.zip	None by default - you have to uncompress and import your properties.zip file manually	Create a folder in Studio
Data XML files For instance: – catalog.xml – store.xml – fulfillment.xml – paymentinfo.xml – currency.xml – shipping.xml – tax.xml	None by default - you have to import them manually	Create a folder in Studio
Document type definition (dtd) dtd files	None by default - you have to import them manually	Create a folder in Studio
Descriptor file sarinfo.xml	None by default - you have to import it manually	Create a folder in Studio

9.5.3 Data flow between WC Studio and SAR

We provide an overview of the data flow between WebSphere Commerce Studio and a SAR file to demonstrate which files are exported to the SAR and which are not (see Table 9-5).

Table 9-5 Data flow between WebSphere Commerce Studio and SAR

Content type and description	Export into SAR	Comment
SAR-INF sarinfo.xml	None by default - you have to export it manually	Define publishing target in Studio
Webapp webapp.zip	Export is done by Studio automatically	No additional action needed
Properties properties.zip	None by default - you have to compress and export your property files manually	Define publishing target in Studio. Compress and add to SAR.
Data XML files For instance: – catalog.xml – store.xml – fulfillment.xml – paymentinfo.xml – currency.xml – shipping.xml – tax.xml	None by default - you have to import them manually	Define publishing target in Studio
Document type definition (dtd) dtd files	None by default -you have to export them manually	Define publishing target in Studio
Descriptor file sarinfo.xml	None by default - you have to export it manually	Define publishing target in Studio

10

Development environment

This chapter provides information for the developer on the tools and environment required to develop and test stores for WebSphere Commerce Suite V5.1

The chapter is organized into the following sections:

- Development environment and tools overview
- Install the development environment
- Configure the development environment
- Tips for development
- Test environments and tools
- Where to find more information

© Copyright IBM Corp. 2001

10.1 Development environment and tools overview

To help developers customize and develop new functionality and new stores, IBM has provided WebSphere Commerce Studio. Commerce Studio ships in two packages, Developer Edition and Professional Developer Edition. The basic difference between the two packages is that the Professional Developer Edition has all the features of the Developer Edition as well being used to develop business logic using Enterprise Java Beans.

The recommended development environment package, and used while creating this redbook, is the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000. Commerce Suite business logic makes extensive use of EJBs. In order to make custom changes to this business logic we need tools that will let us develop and test EJBs; the best tool for this task is IBM VisualAge for Java V3.5, Enterprise Edition (VAJ). This version of VAJ is included in the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 package. VAJ also includes the integrated WebSphere Test Environment (WTE) allowing developers to run Commerce Suite function code within VAJ without exiting the application.

When developing a store for WebSphere Commerce Suite V5.1 it is important to understand the level of customization needed and the development tools provided with WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

This section provides a description of the following development tools:

- Development tools for customization needs
- WebSphere Commerce Studio
  - WebSphere Studio V3.5, Advanced Edition
  - WebSphere Commerce Studio V5.1 extensions
  - VisualAge for Java V3.5, Enterprise Edition
- Other development tools
  - XML editor
  - Store Services
  - IBM PerfectPhoto

10.1.1 Development tools for customization needs

As described in Chapter 2, “Skills planning” on page 15, there are different levels of skills needed for customization of a store. Different tools are required to implement different types and levels of customization. We have classified the development tools into the following categories:

- Basic customization tools
- Intermediate customization tools
- Advanced customization tools

Basic customization tools

Basic customization often requires creating or modifying existing JSPs. WebSphere Studio plus the Commerce extensions are used for JSP development.

In WCS V5.1, you will need an XML editor to modify and create your store catalog data files for such tasks as adding products or product groups.The IBM XML Tool is included on the WebSphere Commerce Studio CD. Other XML editors may be used as well, such as XML Spy. As an alternative to managing your catalog data via XML files, IBM sells a separate product called WebSphere Catalog Manager that provides a GUI front-end for managing your catalog. Information on WebSphere Catalog Manager can be found at:

`http://www.ibm.com/software/webservers/commerce/catalogmanager/`

Tools such as Store Services can be used to modify basic informational settings for your running store. Depending on your graphic customization needs IBM PerfectPhoto can be used for creating or modifying images.

Intermediate customization tools

Intermediate customization tools includes basic customization tools. In addition, VisualAge for Java is used to create custom WCS commands. The use of the WebSphere Test Environment to debug your WCS commands written in Java is also very useful.

Advanced customization tools

Advanced customization includes intermediate customization tools. In addition, VisualAge for Java is used to create custom EJBs.

10.1.2 WebSphere Commerce Studio

WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 contains the majority of the tools required to develop stores for WCS V5.1 Pro Edition, including WebSphere Studio V3.5.1, WebSphere Commerce Studio extensions V5.1, and VisualAge for Java V3.5 Enterprise Edition. This development suite is available for the Windows NT and Windows 2000 platforms. Stores developed with this development suite can be deployed to any WebSphere Commerce Suite V5.1 platform (for example Windows NT, Windows 2000, AIX, Solaris).

We will refer to WebSphere Commerce Studio as Commerce Studio throughout the redbook. In this chapter will explain how to install and configure Commerce Studio. In subsequent chapters, we will provide instructions for using Commerce Studio to import a SAR, customize a store, export a SAR, and deploy a store.

WebSphere Studio V3.5, Advanced Edition

This tool provides visual layout of dynamic Web pages, Studio supports JSPs, full HTML, JavaScript and DHTML, uses wizards (for generating database-driven pages), and updates and corrects links automatically when content changes. WebSphere Studio allows developers to integrate their favorite content creation tools, and provides local and remote debugging with the industry's first JSP debugger.

Some common uses of WebSphere Studio include the following:

- Used to create and modify JSPs and static HTML.
- Used as a repository for all Web assets, including JSPs, HTML, images, Java classes developed in VAJ.
- Used during development to publish Web assets to WCS V5.1 runtime environment.

For information on how to use and program in WebSphere Studio, refer to the following redbooks, which can be found at http://www.redbooks.ibm.com:

- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755

WebSphere Commerce Studio V5.1 extensions

The WebSphere Commerce Suite Studio extension tools allow you to quickly and easily edit Web assets, for instance using Page Designer, then either export them back to the original store archive on the server, or publish them to a running store, or both.

WebSphere Commerce Studio adds the following extensions to WebSphere Studio:

- Store Archive Tools
  - Store Archive template

    This allows you to create a Studio project based on a store archive template. When you create a store archive project, you can import certain file assets from a store archive into WebSphere Commerce Studio.
  - Import Store Archive

    This allows you to import certain file assets from a store archive into a Studio project based on a store archive.
  - Export Store Archive

    This allows you to export file assets from WebSphere Studio to a store archive on the WebSphere Commerce Server.
  - Store Archive Publish Preferences

    Use the Store Archive Publish Preferences to select your preferences, when you publish the file assets from WebSphere Studio to a running store.

Note: We recommend that the store archive publish preference is set to Always update the store archive.

- Blaze Advisor

  The Blaze Advisor is the development component for creating personalization rules (cross-sells, up-sell, etc.) for the Blaze Advisor Rules Server to implement personalization for your store.

VisualAge for Java V3.5, Enterprise Edition

IBM VisualAge for Java V3.5, Enterprise Edition is used to develop and customize WCS commands (Java classes) and EJBs. We refer to IBM VisualAge for Java V3.5, Enterprise Edition as VAJ throughout the redbook.

For information on how to use and program in VisualAge for Java, refer to the following Redbooks, which can be found at http://www.redbooks.ibm.com:

- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *Programming with VisualAge for Java V3.5*, SG24-5264
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755

10.1.3 Other development tools

This section describes development tools used for developing a store that are not installed as part of the WebSphere Commerce Studio, but are included with WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

XML editor

WCS V5.1 makes use of XML in many places within the store, catalog data population, and WCS configuration. An XML editor is an essential part of developing a store. While writing this redbook, we used the IBM XML Tool and XML Spy to edit and create XML files.

- IBM XML Tools

  The IBM XML Tools package is an early technology release for providing XML tooling in the Application Framework for e-business. It is not intended for production use. The IBM XML Tools will be enhanced via a series of ongoing updates. Watch alphaWorks for new updates. This is your chance to let us know whether these tools are useful to you and which features you would like to see added in the final deliverables.

  This tool can be found in the xmltools directory of the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 CD.

- XML Spy

  This is an integrated development environment (IDE) for markup language that has all the major aspects of XML, including XML editing and validation, schema/DTD editing and validation, and XSL editing and transformation.

  A 30-day trial version of XML Spy, with instructions to register for the product for a fee, can be found at: `http://download.cnet.com/`.

Store Services

After developing the store in Commerce Studio, it is desirable to export your store as a SAR file that can be deployed several ways. Store Services is a tool that can be used to publish (deploy) your store to the WebSphere Commerce Suite runtime environment. In addition, settings such as the store profile, shipping, and tax information can be changed on the published store using Store Services.

For more detailed information on Store Services, refer to the online help and *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*.

IBM PerfectPhoto

IBM PerfectPhoto is a complete digital image software package that allows you to manage, create, enhance, touch up, and print images of any kind (images taken with a digital camera, scanned images, bitmap images, or others) in one super-simple and easy-to-use application.

This tool can be used for product display and group display images as well as any other images for your store. IBM PerfectPhoto CD is included in the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 package.

More detailed information on IBM PerfectPhoto can be found at:

http://www.ibm.com/jp/esbu/E/perfectphoto/index.html

IBM HotMedia

HotMedia is a Java applet technology for placing rich interactive media on Web pages quickly and easily. It supports the industry's widest array of media types including 3-D, panoramas, multi-track animations and streaming audio and video. HotMedia includes a toolkit that consists of a simple assembly tool and a set of supporting utilities for authoring HotMedia files. It provides Java applet players that deliver HotMedia objects to any Web browser without the need for plug-ins or special servers, and using today's network bandwidths. Content creators can easily author HotMedia files and seamlessly add a multitude of rich, interactive media effects to any Web page.

This tool can be used to enhance the content display pages of your store. IBM Media is installed by default by the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 install.

More detailed information on IBM HotMedia can be found at:

http://www.ibm.com/software/net.media/

Store archive

The store archive or SAR is a compressed file that contains all the assets to create a store. The outputs of the development of a store are packaged into a SAR file. This not a tool in itself but a packaged unit that is used to publish your store.

For more detailed information on the Store Archive refer to the following:

- Chapter 9, “Store archive overview” on page 271
- Chapter 12, “Creating a store archive (SAR) for deployment” on page 355
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1* product guide found on the WebSphere Commerce Suite V5.1 CD.

10.1.4 Methods of installing the development environment

There are several methods for installing and configuring the WebSphere Commerce Studio:

- WebSphere Commerce Studio - umbrella install

  The procedure installs and configures WebSphere Commerce Studio and VisualAge for Java by using the product-supplied umbrella-like install. If you are not familiar with these tools, we recommend that you use this method of installation. This method of installation works well in an environment that is a clean install (no previously existing components).

  For detailed instruction on installing, refer to the *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000* found in the /docs/<locale> directory of the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

- WebSphere Commerce Studio - manual component install

  This procedure installs and configures WebSphere Commerce Studio components manually. The configuration of VAJ is done manually. This method of installation is available for developers who may already have IBM VisualAge for Java V3.5, Enterprise Edition installed.

  Note: This chapter describes a detailed procedure for installing the development environment, which includes WebSphere Commerce Studio.

- Shared development environment - manual distributed install

  This procedure involves setting up a shared development environment. This shared environment may have multiple WCS instances on a shared WCS runtime environment server.

  This is the ideal development environment, because developers can work together on a single body of code. It makes the best use of the version and source control available in WebSphere Studio and VisualAge for Java. It is, however, undocumented and complex to set up. This environment is beyond the scope of this redbook.

  Another benefit of a share development environment is to offload resources to a common system. This kind of environment can accommodate development shops that may be hardware resource constrained, as described by the requirements of the WebSphere Commerce Studio. The general instructions for this procedure are to install the minimal components of WCS to meet the Commerce Studio install prereq check. The objective is to install Commerce Studio, VAJ and minimal WCS support on your development system. Next, we

will create a separate WCS server with multiple WCS instances for the runtime portion of the development environment, thus offloading the memory requirements of WCS V5.1 to this server and freeing up memory needed on the Commerce Studio/VAJ development system.

10.2 Install the development environment

This section describes the manual procedure for installing, configuring and verifying the development environment. This approach provides guidance for developers who may have already installed IBM WebSphere Studio V3.5.2, Advanced Edition, and IBM VisualAge for Java V3.5, Enterprise Edition + VAJ Fixpack 2. In addition, this procedure provides verification steps to assist the developer in the event that something does not install properly.

We have organized the installation of the development environment into the following phases:

- Development environment hardware and software
- Development environment install prerequisites
- Install WebSphere Studio
- Install VisualAge for Java
- Install WebSphere Commerce Studio
- Install XML editor
- Installing other development tools

10.2.1 Development environment hardware and software

This section defines the hardware and software used within our development environment.

Hardware prerequisites

For product guidelines on the required hardware and software, refer to the *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000*

If you intend to run WebSphere Commerce Suite V5.1, WebSphere Commerce Studio V5.1, and VisualAge for Java V3.5 EE on the same system, we recommend 1 GB of RAM and a minimum of 733 MHz CPU. The minimum for this environment is 512 MB of RAM.

Hardware used in our development environment

This section describes the hardware used within our development environment for Windows NT and Windows 2000 for a 1-tier configuration.

We used two different types of systems within our development environment:

- IBM xSeries Server 230 (8658)
  - 1 GHz PIII CPU
  - 1 GB RAM
  - 18 GB hard disk (installed all software on 5 GB partition)
  - 1 Ethernet (built-in)
- IBM PC 300PL (6565)
  - 733 GHz PIII CPU
  - 512 MB RAM
  - 20 GB hard disk ((installed all software on 5 GB partition)
  - 1 IBM Etherjet 100/10

Software used in our development environment

We used the following software for our development environment:

- Microsoft Windows operating systems:
  - Windows NT V4.0 Server + Service Pack 6a - 128-bit encryption

    or
  - Windows 2000 Server + Service Pack 1 - 128-bit encryption
- Microsoft Internet Explorer V5.5 + Service Pack 1
- Runtime environment components:
  - IBM HTTP Server V1.3.12
  - IBM DB2 Universal Database V7.1, Enterprise Edition for Windows + DB2 V7.1 Fixpack 2a
  - IBM WebSphere Application Server V3.5, Advanced Edition + WAS V3.5 Fixpack 2 + WAS V3.5.2 E-fixes for WCS V5.1
  - IBM WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000
- Development components:
  - IBM WebSphere Studio V3.5.2, Advanced Edition
  - IBM VisualAge for Java V3.5, Enterprise Edition + VAJ V3.5 Patch 2

- IBM WebSphere Commerce Studio V5.1 (extensions)
- XML Editor (IBM XML Tools or XML Spy)

10.2.2 Development environment install prerequisites

Prior to the installation of development components used in WCS development you must complete the following steps:

- Install Microsoft Internet Explorer V5.5 or later

  The Web-based administration tools for WebSphere Commerce Suite V5.1 require the use of Internet Explorer V5.5 or later. This browser can be downloaded from http://www.microsoft.com/. We also installed Service Pack 1, and 128-bit security support.

- Install WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000.

  For detailed instructions, refer to Chapter 6, “WCS runtime for Windows NT and 2000: DB2 and IHS” on page 87.

- Create a WCS instance

  For detailed instructions, refer to Chapter 6, “WCS runtime for Windows NT and 2000: DB2 and IHS” on page 87.

- Publish the InFashion sample store.

  For detailed instructions, refer to 6.7, “Deploy the sample store” on page 115.

Restriction: When creating a store archive using the InFashion sample store, it is imperative that you call the store InFashion. The store must be published with the name InFashion for the WebSphere Commerce Studio configuration to work properly.

We discovered that the configvaj.bat utility used to configure Commerce Studio and VAJ is hard coded for this store name. This utility will be used in subsequent steps.

10.2.3 Install WebSphere Studio

This section provides detailed instructions for installing, configuring, and verifying IBM WebSphere Studio V3.5.2, Advanced Edition.

WebSphere Studio install

To install the IBM WebSphere Studio V3.5.2, Advanced Edition, complete the following steps:

1. Insert the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 CD into the CD-ROM drive.
2. Using Windows Explorer, switch to the WebSphereStudio35 directory on the CD, and double-click **Setup** to start the install.
3. When the Welcome window appears, click **Next**.
4. When the Software License Agreement window appears, read the agreement and if you agree to the terms, then click **Yes** to continue.
5. When the Choose Destination Location window appears, click **Browse**.
6. When the Choose Folder window appears, enter the following path: `c:\ibm\wastudio.` Click **OK**.
7. When the Setup window appears with a message `The folder does not exist, would you like to create it?`, click **Yes**.
8. Click **Next** to proceed.
9. When the Select Components window appears, check **IBM WebSphere Studio V3.5.2** (default), and then click **Next**.

   For the purposes of this redbook, the other options are not required.
10. When the Select Program Folder window appears, accept the default folder (IBM WebSphere), and then click **Next**.
11. When the Start Copying Files window appears, click **Next**.
12. When the Setup Complete window appears, select **Yes, I want to restart my computer**, and then click **Finish**.

WebSphere Studio install verification

To verify the WebSphere Studio installation, complete the following steps:

1. Start WebSphere Studio by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Studio 3.5** -> **IBM WebSphere Studio V3.5**.
2. The first time you start Studio, or if you do not have any projects, the Welcome to IBM WebSphere Studio window appears. Select **Create a new project using**, and from the pull-down select **Default Template** (default), and then click **OK**.
3. When the New Project window appears, enter the Project Name: `test project`, and then click **OK**.
4. If you can create a new project, WebSphere Studio has been installed successfully. Close WebSphere Studio.

The IBM WebSphere Studio V3.5.2, Advanced Edition installation is now complete.

10.2.4 Install VisualAge for Java

This section provides detailed installation and verification instructions for IBM VisualAge for Java V3.5, Enterprise Edition, and VisualAge for Java V3.5 Fixpack 2.

The installation of VisualAge for Java is organized into the following sections:

- VisualAge for Java V3.5 install considerations
- VisualAge for Java V3.5 install
- VisualAge for Java V3.5 Patch 2 install
- VisualAge for Java V3.5.2 verification

VisualAge for Java V3.5 install considerations

IBM VisualAge for Java V3.5, Enterprise Edition has the following installation considerations:

- You must have at least 20 MB free on your Windows system drive, and your environment variable TEMP or TMP must point to a valid temporary directory with at least 6 MB free.
- The CLASSPATH variable must be less than 469 characters. VisualAge for Java inserts approximately 27 characters into the CLASSPATH variable during installation. These numbers are based on the assumption that the target directory is 10 characters long (for example, IBMVJava35).

VisualAge for Java V3.5 install

To install IBM VisualAge for Java V3.5, Enterprise Edition, complete the following steps:

1. Insert the IBM VisualAge for Java V3.5, Enterprise Edition CD 1 into the CD-ROM drive.
2. Using Windows Explorer from the root of the CD, double-click **Setup** to start the install.
3. When the IBM VisualAge for Java Install window appears, click **Install Products** -> **Install VisualAge for Java**.
4. When the Choose Setup Language window appears, select your national language from the drop-down menu (English United States is the default) and click **OK**.
5. When the Welcome window appears, click **Next**.
6. When the Software License Agreement window appears, read the agreement and if you agree to the terms, then select **I accept the terms in the license agreement**, and then click **Next**.

7. When Setup Type window appears, select **Custom by Scenario**, and then click **Next**.
8. When prompted for the selection options, select the following checkboxes and then click **OK**.
   - Create server-side Web applications
   - Create applications that access relational data
   - Create applications using enterprise systems and middleware

Note: If you are developing stores for AS/400 or OS/390, you should select the toolkits for these platforms. If not, you can always go back and add this support if needed in the future.

9. When the Edit Features window appears, click **Change** next to Install to.
10. When the Change Current Destination Folder appears, enter the folder name: `c:\ibm\vaj.` Click **OK**.
11. Click **Next** to proceed.
12. When the Location of the Repository window appears, select **Local** (default), and then click **Next**.
13. When the Ready to Install the Program window appears, click **Install**.

    You should see a window Installing VisualAge for Java for Windows with the status indicator.

Note: The installation of IBM VisualAge for Java V3.5, Enterprise Edition takes approximately 20-30 minutes depending on your system specs.

14. When the InstallShield Wizard Completed window appears, click **Finish**.
15. When the IBM VisualAge for Java Install window appears, click **Exit**.
16. You must restart your system for the configuration changes made to VisualAge for Java.

Note: If your installation fails, refer to the instmigr.html file, which can be found in the root directory of the CD.

For more detailed instructions, refer to the *Installation and Migration Guide for VisualAge for Java V3.5.*

VisualAge for Java V3.5 Patch 2 install

In order to import the WebSphere Commerce Suite V5.1 beans into VisualAge for Java, you must install the IBM VisualAge for Java V3.5, Enterprise Edition V3.5 Patch 2.

To install IBM VisualAge for Java V3.5, Enterprise Edition V3.5 Patch 2, complete the following steps:

1. Insert the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 CD into the CD-ROM drive.
2. Using Windows Explorer, change to the vajfix directory from the root of the CD, and then double-click **Setup** to start the install.
3. When the Choose Setup Language window appears, select your national language from the drop-down menu (English is the default) and click **OK**.
4. The readme.txt file will appear; read the contents of this file and then close it.
5. When the Welcome window appears, click **Next**.
6. The Component Update Results window appears, review the summary and then click **Next**.
7. When the Setup Complete window appears, click **Finish**.
8. After the IBM VisualAge for Java V3.5, Enterprise Edition Patch 2 has been installed, restart your system.

VisualAge for Java V3.5.2 verification

After the installation of VAJ V3.5 and Patch 2, we need to verify that VAJ is working properly by completing the following steps:

1. Start VisualAge for Java by clicking **Start** -> **Programs** -> **IBM VisualAge for Java for Windows V3.5** -> **IBM VisualAge for Java**.
2. The first time you start VisualAge for Java, or if you do not have a repository, you will be prompted to enter the network login name for the Administrator. We entered `admin` and then clicked **OK**.
3. If you can open the Workbench environment, IBM VisualAge for Java V3.5, Enterprise Edition has been successfully installed.
4. Close VisualAge for Java.

The VisualAge for Java installation is now complete.

10.2.5 Install WebSphere Commerce Studio

This section provides detailed installation and verification instructions for WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000.

The installation of WebSphere Commerce Studio is organized into the following sections:

- WebSphere Commerce Studio install prerequisites
- WebSphere Commerce Studio install

WebSphere Commerce Studio install prerequisites

Prior to installing WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000, ensure that you have met the following prerequisites:

- Ensure you are logged in as a Windows user ID with administrator authority.
- Ensure your system meets the install hardware and operating system requirements defined in 10.2.1, “Development environment hardware and software” on page 291.
- Ensure you have installed the Microsoft Internet Explorer V5.5 or higher, as defined in 10.2.1, “Development environment hardware and software” on page 291.
- Ensure your system meets the pre-installation software requirements for the WCS V5.1 runtime environment and WCS instance creation, defined in 10.2.2, “Development environment install prerequisites” on page 293.
- Ensure you have published the InFashion sample store with the store archive name InFashion, defined in 10.2.2, “Development environment install prerequisites” on page 293.
- If you intend to use IBM VisualAge for Java V3.5, Enterprise Edition, it must be installed with Patch 2 prior to the WebSphere Commerce Studio install.

WebSphere Commerce Studio install

To install WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000, complete the following steps:

1. Insert the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 CD into the CD-ROM drive.
2. Using Windows Explorer, from the root of the CD double-click **Setup** to start the install.
3. When the Choose Setup Language window appears, select your national language from the drop-down menu (English is the default) and click **OK**.

4. When the Welcome window appears, click **Next**.
5. Review the content of the Software License Agreement window, and if you accept the conditions, click **Accept** to continue.
6. When the Installation Options window appears, notice that setup will install the Store Archive Tools and Blaze Adviser (not deselectable). Click **Next** to continue.

   For the purposes of this redbook, we did not install any of the remaining options.
7. When the Choose Destination Location window, click **Browse**.
8. When the Choose Folder window appears, enter the following and then click **OK.**

   – Path: `c:\ibm\wcstudio`
9. When the Setup window appears with a message `The folder does not exist, would you like to create it?`, click **Yes**.
10. Click **Next** to proceed.
11. When the Select Program Folder window appears, accept the default (IBM WebSphere Commerce Studio), and click **Next**.
12. When the Summary window appears, review the current settings, and then click **Next** to start copying files.
13. When you are prompted with the message `Installation will now add registry entries to WebSphere Studio. Do you want to continue with this update?`, click **Yes**.
14. When the Setup Complete window appears, select **Yes, I want to restart my computer now**, and then click **Finish** to restart your system.

10.2.6 Install XML editor

WCS V5.1 makes use of XML in many places within the store, catalog data population, and WCS configuration. An XML editor is an essential part of developing a store. While writing this redbook, we used the IBM XML Tool and XML Spy to edit and create XML files.

IBM XML Tools

The IBM XML Tools can be found in the xmltools directory of the WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 CD. Run setup to install IBM XML Tools. We installed it to the c:\ibm\xmltools directory.

XML Spy

A 30-day trial version of XML Spy, with instructions to register for the product for a fee, can be found at: http://download.cnet.com/.

10.2.7 Installing other development tools

The installation of the other development tools mentioned in 10.1, “Development environment and tools overview” on page 284 is optional for the purposes of this redbook.

For installation instructions, refer to the product installation guides and the *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000*.

10.3 Configure the development environment

After the installation of the development environment, WebSphere Studio and VisualAge for Java need to be configured to develop customized WebSphere Commerce Suite code.

10.3.1 Configure WebSphere Studio

This section provides configuration instructions for WebSphere Studio after the installation of the tools.

Configure the JDK preferences

If you installed WebSphere Studio prior to installing WebSphere Commerce Studio, and you are using VisualAge for Java, complete the following steps to configure the JDK settings in WebSphere Studio:

1. Start WebSphere Studio by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Studio 3.5** -> **IBM WebSphere Studio V3.5**.
2. From the WebSphere Studio menu bar, select **Tools** -> **Preferences**.
3. Select the **Java** tab in the Preferences window.
4. Ensure that the <CommerceStudio_install_path>\lib\wcsdatabeans.dev.jar is in the following classpaths as seen in Figure 10-1:
   - JDK Version 1.1.x, classpath (1.1.x):

     <CommerceStudio_install_path>\lib\wcsdatabeans.dev.jar
   - JDK Version 1.2.x, classpath (1.2.x):

     <CommerceStudio_install_path>\lib\wcsdatabeans.dev.jar

If the file is not listed in the classpath, add it to each and then click **OK**.

For example, add c:\ibm\wcstudio\lib\wcsdatabeans.dev.jar, where wcstudio is the Commerce Studio install path.

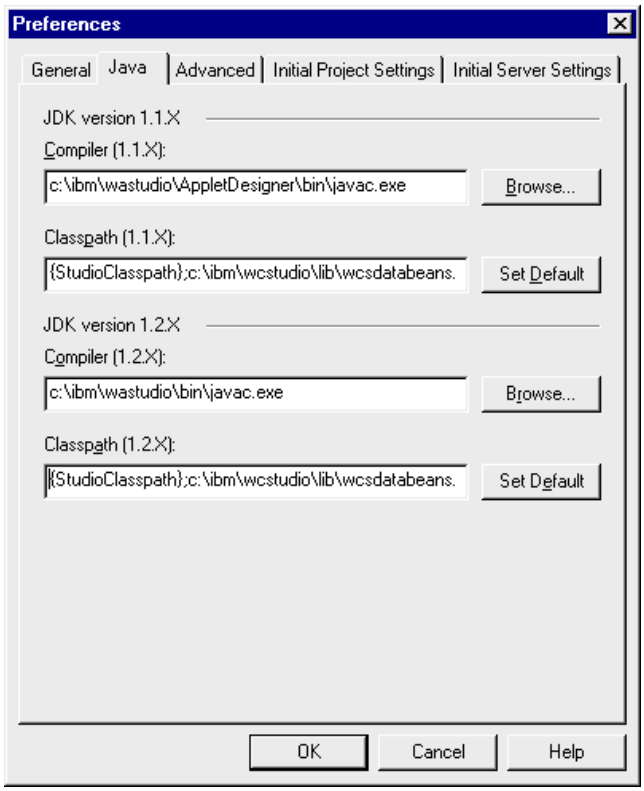


Figure 10-1 Configure the JDK classpath in Studio

Configure the store archive publish preferences

To modify the store archive publish preferences, complete the following steps:

1. Start WebSphere Studio by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Studio 3.5** -> **IBM WebSphere Studio V3.5**.
2. Select your project folder. For example, we created a project called test project.
3. From the WebSphere Studio menu bar, select **Tools** -> **Wizards** -> **Store Archive Publish Preferences**.
4. When the Store Archive Publish Preferences window appears, ensure that **Always update the store archive** is selected (default). This will update the store archive with the Web asset files during publishing to a running store.

Registering tools (XML editor)

If you installed XML Spy and want to edit files directly from Studio, you can register XML Spy to be the default editor for XML files.

1. Start WebSphere Studio.
2. Click **Tools** -> **Tools Registration** from the menu bar.
3. When the Tools Registration window appears, enter `xml` in the file extension file, and then click **Edit**. Notice that Notepad is the default editor.
4. When the Editor object type window appears, select the **Editing Applications** tab.

   XML Spy does not appear in the Registered applications list.
5. Click **Browse** to find the XML Spy executable.
6. Select **XMLspy.exe** from the directory that you installed XML Spy, and then click **Open**.
7. From the Editors pane, select **XMLspy.exe** and then click **Set as Default**.
8. Click **OK**.
9. Repeat this process for other file extensions and tools.

10.3.2 Configure VisualAge for Java

This section describes how to configure VisualAge for Java after the development environment install. This configuration only needs to be performed one time on your development system. In order to develop customized WebSphere Commerce Suite Java code,

We have organized the configuration of VisualAge for Java into the following phases:

- VisualAge for Java configuration prerequisites
- Add features to VisualAge for Java
- Import wcs5101.dat and patches into VAJ WTE
- Increase memory available to VAJ
- Run the configvaj.bat script
- Configure the EJB and PNS servers
- Start the EJB server
- Configure rules services (optional)
- Import rules resources
- Start the Servlet Engine

VisualAge for Java configuration prerequisites

- Prior to the VAJ configuration, you must have installed VAJ before installing WebSphere Commerce Studio.
- Ensure that you stop the WebSphere Application Server Windows service, IBM WS AdminServer. This is needed due to conflicting ports used by the WebSphere Application Server and VisualAge for Java WebSphere Test Environment.

Add features to VisualAge for Java

To add features required for WCS development to VisualAge for Java, complete the following steps:

1. Start VisualAge for Java by clicking **Start** -> **Programs** -> **IBM VisualAge for Java for Windows V3.5** -> **IBM VisualAge for Java**.

Note: The first time you start VisualAge for Java, or if you do not have a repository, you will be prompted, Enter the network login name for the Administrator. For example, we entered admin and then clicked **OK**.

2. When the Welcome window appears, uncheck **Show this window at startup**, and then click **Close**.
3. Press F2 to open the Quick Start window.
4. Select **Features** -> **Add Feature**, and then click **OK**.
5. When the Selection Required window appears, select the following features:
   - IBM EJB Development Environment 3.5
   - IBM Common Connector Framework 3.5
   - IBM Java Record Library 3.5.0.1
   - Click **OK**

Note: This procedure takes approximately 5-10 minutes depending on your system specs.

Press the Ctrl key to select more than one feature.

Import wcs5101.dat and patches into VAJ WTE

To import the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 repository (wcs5101.dat), and WAS V3.5 E-fixes into the VisualAge for Java WebSphere Test Environment, complete the following steps:

1. Insert the WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 CD into the CD-ROM drive. This CD contains the WAS V3.5 E-fixes, and the WebSphere Commerce Suite V5.1 repository for VAJ.
2. From the VisualAge for Java Workspace, ensure that the workspace owner is Administrator by completing the following steps:
   a. From the menu bar, select **Workspace** -> **Change Workspace Owner**.
   b. When prompted to select the workspace owner, select **Administrator**, and then click **OK**.
3. With the Project tab selected, select the **IBM WebSphere Test Environment project**. Right-click **Manage** -> **Create Open Edition**.

   When the process has completed, the Create Open Edition option will be unavailable.
4. Import 85699_350.jar and 88452_350.jar from <cd_drive>:\wasfix\wasefix\was 3.5.0 efixes into the IBM WebSphere Test Environment project by performing the following steps:
   a. From the File menu select **Import**.
   b. When the Import SmartGuide window appears, select **Jar file** and click **Next**.
   c. Click **Browse** next to Filename, and then select <cd_drive>:\wasfix\wasefix\was 3.5.0 efixes\85699_350.jar as the file name, and click **Open**.
   d. Click **Finish**.
   e. When prompted `Should creation of an edition of this package be attempted?`, click **Yes to all**.
5. Repeat previous step to import 88452_350.jar.
6. To import the wcs5101.dat repository, complete the following steps:
   a. From the File menu, select **Import**.
   b. When the Import SmartGuide window appears, select **Repository**, and then click **Next**.
   a. When the Import from another repository window appears, select **Local repository**.
   b. In the Repository name field, enter

      `<cd_drive>:/repository/WCS5101.dat`
   c. Select **Projects**, and click **Details**.
   d. Select the following projects:
      - IBM WCS Commerce Server

- WCS Enterprise Beans
- In the right-hand pane, select the available versions, then click **OK**.

e. Ensure that the **Add most recent project edition to workspace** check box is selected.

f. Click **Finish** to begin importing.

Importing the WCS5101.dat repository takes approximately 10-15 minutes depending on your system specs.

Note: During the import, many errors related to deprecated classes will occur. These errors will not affect the functionality of the product.

7. 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**.
8. Save your workspace by selecting **Save Workspace** from the File menu.

Increase memory available to VAJ

During development, you may encounter exceptions indicating that the system is out of memory, even when system resources are available. To fix this problem, modify the virtual memory allocation for VisualAge for Java by completing the following steps:

1. Close VAJ.
2. Change to the directory of the VAJ ide.ini file found in <vaj_install_path>\ide\program\.

   For example:

   ```
   c:\ibm\vaj\ide\program
   ```
3. Modify the ide.ini file with a text editor.

   Insert the following at the bottom of the file:

   ```
   [VM options]
   maximumMemoryLimit=536870912
   ```
4. The next time you start VAJ, the changes will be in effect.

Run the configvaj.bat script

Once you have imported the required files into VisualAge for Java (done in previous step), you must run the configvaj.bat script that performs configuration steps related to using the WebSphere Test Environment, within VisualAge for Java.

Run the configvaj.bat s script, by completing the following steps:

1. Close VisualAge for Java.
2. Temporarily modify your PATH system environment variable to include the Java in the path, by typing the following at a command prompt:

   ```
   set PATH=%PATH%;<drive>:\<was_install_path>\jdk\bin
   ```

   Where <was_install_path> is the WebSphere Application Server install path.
3. Test that the Java is working in the temporary path environment variable, by typing `java` at the command prompt. It should return a Java usage message.
4. Modify the java.security file as follows:
   a. Change to the <was_install_path>\jdk\jre\lib\security directory.
   b. Back up java.security to java.security.bak
   c. Search for the following line in a text editor (for example, Wordpad):

      ```
      security.provider.1=sun.security.provider.Sun
      ```

   d. Add the following line if it is not already there from the WCS V5.1 runtime configuration:

      ```
      security.provider.2=com.ibm.crypto.provider.IBMJCE
      ```

   e. Delete the following line if found in the file:

      ```
      security.provider.2=com.sun.crypto.provider.SunJCE
      ```

   f. Save the file.

   Note: Ensure there are no trailing spaces.

5. Modify the configvaj.bat to include the correct paths by completing the following steps:
   a. Change to the <drive>:\<CommerceStudio_install_path>\bin directory.
   b. Back up configvaj.bat to configvaj.bat.bak.
   c. Modify configvaj.bat file with a text editor.
   d. Search for the following lines:

      ```
      set JDK_PATH=[JDKPath]
      set WCS_PATH=[WCSPath]
      set JDBC_DRIVER=[JDBCDriver]
      ```

   e. Replace the variables in [ ] with the correct paths for your installation.

      For example, we modified the paths as follows for our environment:

      ```
      set JDK_PATH=c:\ibm\was\jdk
      set WCS_PATH=c:\ibm\wcs
      ```

```
set JDBC_DRIVER=c:\ibm\sqllib\java\db2java.zip
```

6. At a command prompt, switch to the <drive>:\<CommerceStudio_install_path>\bin directory.
7. Run the **configvaj.bat** command as follows to set up the configuration file for the VisualAge for Java test environment:

```
> configvaj admin admin c:\ibm\sqllib c:\ibm\wcs c:\ibm\vaj
c:\ibm\was\jdk c:\ibm\sqllib\java\db2java.zip wcs InFashion
```

Syntax:

```
> configvaj <dbuser> <dbpassword> <db_install_path> <wcs_install_path>
<vaj_install_path> <jdk_install_path> <jdbc_path_jdbcdriver>
<instanceName> <storeName>
```

Note: If any of the paths contain spaces, enclose the path in double quotation marks(“ “).

The execution of the configvaj.bat script runs approximately 5-10 minuets depending on your system specs.

8. Review the <drive>:\<CommerceStudio_install_path>\classes\configvaj.log file for errors and successes. We have included a sample configvaj.log in Example 10-1 to show a successful execution of configvaj.bat.

Example 10-1 Sample configvaj.log

```
Hostname = m23vnx58
DBUser = admin
DBPswd = rdquwFe7Pxc=
DB2BinPath = c:\ibm\sqllib\bin
WCSPath = c:\ibm\wcs
VAJPath = c:\ibm\vaj
JDKPath = c:\ibm\was\jdk
JDBCDriver = c:\ibm\sqllib\java\db2java.zip
InstanceName = wcs
StoreName = InFashion
Modifying java.security
Modifying java.security
Modifying Resource Class Path
Copying files from c:\ibm\wcs/web
Copying files from c:\ibm\wcs/properties
Copying store files
Modifying default_app.webapp file
default_app.webapp file is already updated
Copying exclude.list file
exclude.list already copied
Program completed successfully
```

Configure the EJB and PNS servers

Configure the EJB server by completing the following steps:

1. Start VisualAge for Java.
2. From the Workspace, click the **EJB** tab.
3. Select the EJBs whose names begin with **WCS** to deploy.

   Press the Ctrl key to select more than one.
4. With the EJBs to deploy highlighted from the previous step, right-click and select **Add To** -> **Server Configuration**.
5. When the EJB Server Configuration window appears, right-click the **EJB Server**, and then select **Properties**. For example, EJB Server (server1).
6. When the Properties for EJB Server window appears, enter the properties values described in Table 10-1.

   For example, we entered the following for DB2:

   – Data Source: `WebSphere Commerce Suite DB2 DataSource wcs`

   – Connection Type: select **<DataSource>** from pull-down

   where <DataSource> is the name of the connection type.

   – Database User ID: `admin`

   – Database Password: `admin`

   – Click **OK**

Table 10-1 EJB Server property description

Property	DB2 value	Oracle value
Data Source	WebSphere Commerce Suite DB2 DataSource <wcs_instance_name> Note: This can be found in the WAS repository via the Administrator's Console.	WebSphere Commerce Suite Oracle DataSource <wcs_instance_name> Note: This can be found in the WAS repository via the Administrator's Console.
Connection Type	<DataSource> Note: this is the actual name.	<DataSource>
Database User ID	<db_user>	<db_user>
Database Password	<db_user_password>	<db_user_password>

7. To create a Persistent Name Server, complete the following steps:

   DB2 users:

   i. Open a DB2 Command Window by clicking **Start** -> **Programs** -> **IBM DB2** -> **Command Window**.

   ii. Type the following DB2 command to create the PNS database:

   ```
   > db2 create db <PNS_db_name>
   ```

   Where <PNS_db_name> is the name of your Persistent Name Server database.

   For example, we entered the following:

   ```
   > db2 create db PNS
   ```

   Oracle users:

   To create a PNS for Oracle, create a tablespace for the PNS user, and a PNS user and password in your Oracle database.

8. In the EJB Server Configuration window, select **Workspace** -> **Tools** -> **WebSphere Test Environment**.

9. When the WebSphere Test Environment Control Center window appears, select **Persistent Name Server**.

10. In the Persistent Name Server window appears, enter the property values described in Table 10-2.

    For example, we entered the following for DB2:

    – Database Driver: select **COM.ibm.db2.jdbc.app.DB2Driver** from the pull-down. The other fields will now be accessible.

    – Database URL: `jdbc:db2:PNS`

    – Database User ID: `admin`

    – Database Password: `admin`

    Click **Apply**.

Table 10-2 Persistent Name Server property description

Property	DB2 value	Oracle value
Database URL	jdbc:db2:<pns_db_name> Note: PNS database created in previous step.	jdbc:oracle:thin@<hostname>:1521:SID
Database Driver	COM.ibm.db2.jdbc.app.DB2Driver	oracle.jdbc.driver.OracleDriver
Database User ID	<pns_db_user>	<pns_user>
Database Password	<pns_db_user_pw>	<pns_user_pw>

11. Click **Start Name Server**.

    If successfully started the last line in the console should be as follows:

    ```
    E Server open for business
    ```

12. Configure your datasource.

    DB2 users:

    – From the WebSphere Test Environment Control Center window, click **DataSource Configuration**.

    – When the DataSource Configuration window appears, click **Add**.

    – When the Add DataSource window appears, enter the property values described in Table 10-3.

      For example, we entered the following:

      - DataSource Name: `WebSphere Commerce Suite DB2 DataSource wcs`
      - Database Driver: select **COM.ibm.db2.jdbc.app.DB2Driver** from pull-down
      - Database URL: `jdbc:db2:wcs`
      - Click **OK**.

Table 10-3 DB2 DataSource Configuration property description

Property	Value
DataSource Name	WebSphere Commerce Suite DB2 DataSource <wcs_instance_name>
Database Driver	COM.ibm.db2.jdbc.app.DB2Driver
Database URL	jdbc:db2:<your_wcs_db_name>

Oracle users:

Your datasource should already be configured. You can confirm this by looking for the following line in your console when you start the PNS server:

```
E resolve :name =jdbc cname = /
```

If this line is not present, click **DataSource Configuration**, then **Add**, and then enter the information as described in Table 10-4.

Table 10-4 Oracle DataSource Configuration property description

Property	Value
DataSource Name	WebSphere Commerce Suite Oracle DataSource <wcs_instance_name>
Database Driver	oracle.jdbc.driver.OracleDriver
Database URL	jdbc:oracle:thin@<hostname>:1521:SID

13. From the WebSphere Test Environment Control Center, select **Persistent Name Server**, and then click **Stop Name Server**. Wait for the process to end.
14. Click **Start Name Server** for the changes to take effect.

    The console should report that the server has successfully started:

    ```
    E Server open for business.
    ```

Start the EJB server

To start the EJB Server, complete the following steps:

1. Select the **EJB** tab, then select **EJB** from the menu bar. Select **Open To** -> **Server Configuration**.
2. Select the **EJB Server Configuration** window, right-click **EJB Server** and select **Start Server**. For example, EJB Server (server1).
3. Wait for several minutes and check your console to see if your server is started properly.

   Select **EJB Server** in the Console and the last message should say:

   ```
   E-Server open for business.
   ```

Configure rules services (optional)

If your Commerce Suite instance uses rules services, you must complete the following steps so that rules services will function in the VAJ WTE:

1. Change to your wcs instance XML directory:

   ```
   <drive>:\<wcs_install_path>\instances\<instance_name>\xml
   ```

For example:

```
c:\ibm\wcs\instances\wcs\xml
```

Where `wcs` is the name of our wcs instance.

2. Modify the <instance_name>.xml file with a text editor. Locate the RuleService tag, modify the section to look as follows:

```
<RuleService display="false"
       name ="Personalization"
       RuleServerEnabled="false" >
<CampaignManagement  display="false"
       CampaignEnabled="false"/>
</RuleServices>
```

Import rules resources

These steps are required even if you are not using rules services, because the Servlet Engine Configuration makes references to these resources upon startup.

1. Start VisualAge for Java.
2. Create a new project to include the resource files by selecting **Add** -> **Project** from the menu.
3. When the Create New Project Named field, enter `Rules Resources` and click **Finish**.
4. To add the rules JAR files, complete the following steps:
   a. Select the newly created project, Rules Resources, and right-click and select **Import**.
   b. When the File Import SmartGuide window appears, select **Jar file** and click **Next**.
   c. In the Filename field, enter the following:

      ```
      c:\ibm\wcs\blaze\AdvSvr31\lib\AdvCommon.jar
      ```

      Where `c:\ibm\wcs` is the WCS install path.
   d. Ensure that only the **Resource** checkbox is checked.
   e. Click **Browse** to specify into which project the JAR file will be imported, and select the **Rules Resources** project.
   f. Select **Overwrite existing files without warning**.
   g. Click **Finish**.
5. Repeat steps to add the following JAR files:

```
c:\ibm\wcs\blaze\AdvSvr31\lib\Advisor.jar
c:\ibm\wcs\blaze\AdvSvr31\lib\AdvisorSvr.jar
c:\ibm\wcs\blaze\AdvIrt31\lib\InnovatorRT.jar
```

Once you have completed these steps, you can start the Servlet Engine.

Start the Servlet Engine

When the EJB server has started, switch back to the WebSphere Test Environment Control Center to start the Servlet Engine, by doing the following:

1. Because VisualAge for Java WebSphere Test Environment does not support SSL, disable SSL mode by completing one of the following:

   Important: Before disabling SSL with a global table-wide update, take note of the settings before this update to enable the database to go back to the secure HTTPS state.

   DB2 users:

   a. Start a DB2 Command window.

   b. Connect to the WCS database as follows:

   ```
   db2 connect to wcs
   ```

   c. Query the viewreg table to find out what is currently set with https=1 (on) and direct the information to a file as follows:

   ```
   db2 select viewname, devicefmt_id, storeent_id from viewreg where
   https=1 > c:\temp\viewregssl.log
   ```

   d. Query the urlreg table to find out what is currently set with https=1 (on) and direct the information to a file as follows:

   ```
   db2 select url, storeent_id from urlreg where https=1 >
   c:\temp\urlregssl.log
   ```

   e. To disable SSL, type the following DB2 commands:

   ```
   db2 connect to <wcs_database>
   db2 update viewreg set https=0
   db2 update urlreg set https=0
   ```

   Note: To enable SSL, use the information in the viewregssl.log and urlregssl.log to change from https=0 (SSL disabled) to https=1 (SSL enabled) for each entry in the log file.

   Oracle users:

   a. Start an SQL Plus window.

   b. Connect to your WebSphere Commerce Suite database with your WebSphere Commerce Suite user name and password.

c. Type the following at the prompt:

```
update viewreg set https=0;
update urlreg set https=0;
```

2. Modify the VAJ default_app.web.

   a. Change to the following directory:

      <vaj_install_path>\ide\project_resources\IBM WebSphereTest Environment\hosts\default_host\default_app\servlets

   b. Add the servlet defined in Example 10-2 to the default_app.webapp file, and modify the following for your store (store name, wcs instance config file and path, hostname):

Example 10-2 Sample servlet for VAJ

```
<servlet>
   <name>InFashion</name>
   <description></description>
   <code>com.ibm.commerce.server.RequestServlet</code>
   <init-parameter>
      <name>configfile</name>
      <value>c:\ibm\wcs\instances\wcs\xml\wcs.xml</value>
   </init-parameter>
   <init-parameter>
      <name>instancename</name>
      <value>m23vnx58</value>
   </init-parameter>
   <autostart>false</autostart>
   <servlet-path>/stores/store_name </servlet-path>
</servlet>
```

      Where `store_name` is the name of your store, and `host_name` is the fully qualified name of your Commerce Suite machine.

3. In the WebSphere Test Environment Control Center, click **Servlet Engine**, then, in the Servers window, click **Edit Classpath**.
4. Click **Select All**.
5. In the Extra class path field, enter `<drive>:\<wcs_install_path>\blaze\license` and click **OK**.
6. Check the **Display trace messages** checkbox, then click **Apply**.
7. Click **Start Servlet Engine**.

Note: If you remove any project from your workspace and put it back in later, you need to add that project back into your Servlet Engine classpath. The WebSphere Test Environment does not include new projects into the Servlet Engine classpath automatically.

8. Verify that the Servlet Engine has started correctly in the Console. You should see the following message:

   ```
   *** Servlet Engine is started ***
   ```

9. To test your environment, open a browser and enter the following URL:

   ```
   http://localhost:8080/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId
   =10001&catalogId=10001&langId=-1
   ```

Note: The storeID and catalogId values may be different for your store.

10.4 Tips for development

We have listed several tips that will reduce the amount of memory used on your development system.

10.4.1 Stop unneeded Windows services

There are several DB2, IBM HTTP, and WCS services that can be stopped to reduce memory used on your development system. Collectively, stopping them can make a difference in the ability of your system to run in real memory versus paging.

Stop DB2 unused services

We have listed the DB2 services that are set to automatic and started after the DB2 installation. To conserve on system memory, we do not start some of the services that are not needed for most WCS applications. The startup type for the service can be set to *automatic, manual,* or *disabled*. Table 10-5 lists the services settings as they are after installation, and our recommended settings.

Table 10-5 DB2 Windows Services

Windows DB2 service name	Service startup type after installation	Service startup recommended setting
DB2 - DB2	automatic	automatic
DB2 - DB2CTLSV	automatic	automatic
DB2 - DB2DASD00	automatic	automatic
DB2 Governor	manual	manual
DB2 JDBC Applet Server	automatic	automatic
DB2 JDBC Applet Server - Control Center	manual	manual
DB2 License Server	automatic	automatic
DB2 Security Server	automatic	manual *
Warehouse logger	automatic	manual
Warehouse server	automatic	manual
* We stopped the DB2 Security Server Windows service within our test environment. If your production runtime environment requires this level of security, set this service to automatic.		

Stop other services (HTTP, WCS)

- IBM HTTP Administration

  When not using this service, we recommend that you stop the service and set it to *manual*.

- IBM WCS Configuration Manager Server

  When not using this service, we recommend that you stop the service and set it to *manual*. This will also eliminate a potential security risk.

10.4.2 Disable Payment Manager

WCS V5.1 requires that Payment Manager V2.2 be installed on the WCS V5.1 server. In order to reduce the memory requirements of the development system, we have devised a procedure to disable the requirement of needing Payment Manager installed on development system to complete order in sample store.

Note: This section assumes that Payment Manager is installed. If Payment Manager is not installed, you *must* complete step 3.

To disable Payment Manager for development system testing, complete the following steps:

1. Ensure that the Payment Engine is not running. This is a separate process that may be running on your machine. Normally, this process will only be running if it has been explicitly invoked. It is not automatic. In order to make sure that it is not running, run the command:

   ```
   <payment_manager_install_dir>/StopIBMPayServer.cmd
   ```

2. Stop the Payment Manager server node in the WebSphere Application Server Administration Console. This can be accomplished as follows:

   a. Open the WebSphere Application Server Administrative Console by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Application Server v3.5** -> **Administrator's Console**.

   b. Expand <your_host_name>.

   c. Right-click the node WebSphere Payment Manager, and select **stop**.

   d. Close the WebSphere Application Server Administrative Console.

3. The default payment behavior of WCS V5.1 is to use Payment Manager for payment processing. Thus, by default payment processing is implemented by Payment Manager commands. The final step in disabling payment processing is to redirect the commands so that they are carried out by non-Payment Manager implementations. The following is an example using a Windows NT environment with a DB2 database.

   a. Connect to your database server.

      Click **Start** -> **IBM DB2** -> **Command Line Processor**.

   b. Connect to the Commerce Suite database. This is the database that was specified during WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 instance creation. For example, if your database was named "mall", you would type the command:

      ```
      db2 => connect to mall
      ```

   c. Update the CMDREG table using SQL as follows:

      ```
      db2 => UPDATE cmdreg SET
      classname='com.ibm.commerce.payment.commands.DoPaymentCmdImpl' WHERE
      interfacename='com.ibm.commerce.payment.commands.DoPaymentCmd'
      AND storeent_id=10001
      db2 => commit
      ```

Note: To revert the implementation to Payment Manager, simply repeat the same process and supply the class name of the Payment Manager class name.

The class name for Payment Manager implementation is:

com.ibm.commerce.payment.commands.DoPaymentMPFCmdImpl.

10.4.3 Starting Payment Manager as a Windows service

For further information, refer to *Install Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2*.

To start the Payment Engine as a Windows NT service for the first time, you must install the Payment Manager NT service:

1. Go to the directory where the Payment Manager is installed.
2. Change directories to the ntservice subdirectory.
3. Type `InstallPaymentManagerNTService` and press Enter.

10.4.4 Cache tips

When developing and modifying files such as JSPs frequently, you will want to test them without having to constantly clear the cache. Disabling the cache may be a good alternative for development purposes. Also, this will cut down on system memory used. This applies to the WCS runtime test environment.

- Disable caching from the WCS Configuration Manager

  This is done to conserve on memory, and avoid testing with cached pages instead of the desired newly developed page.

- Create shortcuts to the cache directories to make it easier to clear cache.

  For example:

  WCS: \<wcs_install_path>\instances\<wcs_instance>\cache

  WAS: \<was_install_path>\xxx

  Browser: \<winnt>\profiles\<user>\cachexxx

10.5 Test environments and tools

In order to test your store application before deployment, you will need a proper test environment. We have divided the WCS test environments into the following:

- VAJ WebSphere Test Environment (WTE) + WCS V5.1

  This environment is crucial when creating business logic in Java. The WTE provides a debug environment for JSPs, servlets, and EJBs.

- WebSphere Commerce Suite V5.1 runtime environment

  This environment is used to test your code in a test environment that is more representative of a customer environment. Additional components such as Websphere Payment Manager, MQSeries, and LDAP can be integrated into this environment for testing purposes.

- Standard PC Web browser (Netscape Navigator, Microsoft Internet Explorer)

10.6 Where to find more information

IBM Redbooks

The following Redbooks can be found at: `http://www.ibm.com/redbooks`:

- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *Programming with VisualAge for Java V3.5*, SG24-5264
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174
- *Patterns for e-business: User-to-Business Patterns for Topology 1 and 2 using WebSphere Advanced Edition*, SG24-5864

IBM WebSphere Commerce Suite product guides

The following WebSphere Commerce Suite V5.1 product guides can be found on the WebSphere Commerce Suite V5.1 Pro Edition CD in the docs/<locale> directory.

- *Fundamentals, IBM WebSphere Commerce Suite V5.1*
- *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition (for specific platform)*
- *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*

IBM WebSphere Commerce Studio product guides

The following WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000 product guide can be found on the WebSphere Commerce Suite V5.1 Pro Edition CD in the docs/<locale> directory:

- *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000*

Online help for WebSphere Commerce Studio

The online help for WebSphere Commerce Studio can be found by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Documentation**.

11

Create and customize a store using Commerce Studio

In this chapter we provide information on store creation and basic store customization by using examples of a Business-to-Consumer (B2C) type of store. By using WebSphere Commerce Studio (Studio, VAJ) we can develop, manage, and publish all store assets from one workbench.

This chapter is organized into the following sections:

- Overview
- Create a store
- Load store assets into Commerce Studio
- Configuring Studio for publishing
- Publishing from Studio
- Basic customization of Web assets
- Where to find more information

© Copyright IBM Corp. 2001

11.1 Overview

This chapter provides simple examples of basic store customizations of store assets such as text, images, static HTML, JSPs, and basic commands. Please keep in mind that this chapter is intended to give you an overview on common *basic* customization techniques.

For detailed information on customization procedures please refer to:

- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174
- *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*

Note: All examples are based on the WebFashion sample store.

11.1.1 Types of stores

WebSphere Commerce Suite V5.1, Pro Edition for Windows NT and Windows 2000 acts as a platform for the following e-businesses and their needs for individually customized store:

- B2C (Business-to-Consumer e-commerce)

  Business-to-Consumer stores employ the B2C model of online retail sales.

  For more information and case studies about B2C, visit IBM's B2C e-commerce solutions Web page on:

  `http://www.ibm.com/software/webservers/commerce/btoc/`

- B2B (Business-to-Business e-commerce)

  Business-to-Business stores employ the B2B model in which two businesses or organizations facilitate their interaction through Web interfaces. Interaction between suppliers and retailers is a common form of this model.

  For more information and case studies about B2B visit IBM's B2B e-commerce solutions Web page on:

  `http://www.ibm.com/software/webservers/commerce/btob/`

- Auctions

  For more detailed information on auctions and how to implement and use them in a WCS V5.1 store, refer to Chapter 14, "Implementing auctions" on page 401.

11.1.2 Approaches to creating a store

There are two alternative approaches to creating a WCS store:

1. Build a store from a demo store

   Use the demo store provided with WCS as a framework and adapt and/or extend the store and its assets to meet your specific business requirements.

   We will provide a sample in this chapter of this approach.

   This concept may also include the user building template stores for various business segments and using them as templates for store creation.

2. Build a store from scratch

   This approach is used when the business requirements for advanced customization require many capabilities beyond the features implemented in the sample store. This approach is beyond the scope of this redbook.

Whatever option you choose to build your own store, WC Studio is the tool to choose when it comes to changing store assets.

11.2 Create a store

This section describes how to create a store from a template.

11.2.1 Create a store template

The first step in store creation is the generation of a template store from which we will create the actual stores to be customized. For the purpose of our examples, we make a copy of the WebFashion store to serve as a template. This is accomplished using the following steps:

1. Download the WebFashion store (WebFashion.zip) from the following URL:

   `http://www.ibm.com/software/webservers/commerce/wcs_pro/downloads.html`

2. Make copies of the following files in the <wcs_install_path>\xml\sar\ directory:

Table 11-1 Back up the original DTD files

Original File:	Backup to:
catalog.dtd	catalog.dtd.bak
command.dtd	command.dtd.bak
store-all.dtd	store-all.dtd.bak

3. Extract the downloaded file, WebFashion.zip directly into the WCS install path (for example, c:\ibm\wcs).

Note: For detailed instructions on installing the WebFashion sample store, refer to the Web Fashion Installation, Configuration, and User documentation. This document should have been unzipped as a PDF file in the directory <wcs_install_path>\samples\stores\WebFashion.

4. Add the WebFashion JAR file to the class path. The JAR contains Java customized commands. This is accomplished as follows:
   a. Start the WebSphere Application Server Administrator's Console.
   b. Expand the **WebSphere Administrative Domain** -> **hostname** node.
   c. Right-click **WebSphere Commerce Server - <instance_name>**.
   d. Select **Stop**. After a few moments, a message stating that it stopped successfully displays.
   e. In the Command line arguments field, append the following text, all on one line, at the end of the class path parameter:

      ```
      <wcs_install_path>\lib\WebFashion.jar
      ```

   f. Click **Apply** to apply the changes.
   g. Right-click **WebSphere Commerce Server - <instance_name>**.
   h. Select **Start**. After a few moments, a message stating that it started successfully displays. You can now close the Administrator's Console.
5. Create a new store owner for WebFashion as follows:
   a. Open a DB2 command line processor by clicking **Start** -> **Programs** -> **IBM DB2** -> **Command Line Processor**
   b. Issue the following commands to create a new owner:

      ```
      db2 => connect to <wcs_database> user <dbuser> using <dbuser_password>
      db2 => insert into member (member_id,type) values (1111, 'O')
      db2 => insert into orgentity (orgentity_id, orgentityname,
      orgentitytype) values (1111,'IBM ITSO', 'O')
      db2 => commit
      ```

      Note: This is the letter O not the number 0.

6. Modify the commit count number of your Commerce Suite instance as follows:
   a. In a text editor or your XML editor, open the file:

      ```
      <wcs_install_path>\instances\<instance_name>\xml\<instance_name>.xml
      ```

b. Under the DevTools tag, set the CommitCount value to 10500.

c. Save and close the file.

7. When using DB2, increase the log file size of your Commerce Suite database. This is necessary in order to publish WebFashion. It is accomplished as follows:

   a. Open the DB2 Control Center by clicking **Start** -> **Programs** -> **IBM DB2** -> **Control Center**.

   b. Expand the **<hostname>** -> **Instances** -> **DB2** -> **Databases** node.

   c. Right-click on your Commerce Suite database (by default, this is named “MALL”).

   d. Select **Configure...**

   e. In the Configure Database window, select the **Logs** tab.

   f. In the Logs tab, change the value of the Log file size so that it is at least 2500 (default 250), and then click **OK**.

   Note: If you do not increase the log file size, you may experience failures publishing a store archive using Store Services.

   g. Exit the Control Center.

8. Create a folder named ITSODEMO for your new store in the <wcs_install_path>\samples\stores directory. For example:

   c:\ibm\wcs\samples\stores\itsodemo

9. Copy the original WebFashion SAR file (WebFashion_en_US_es_ES.sar) in the WebFashion directory to your store directory with the name of the SAR file name as your store. For example:

   From:

   c:\ibm\wcs\samples\stores\WebFashion\WebFashion_en_US_es_ES.sar

   To:

   c:\ibm\wcs\samples\stores\itsodemo\ITSODEMO.sar

10. Copy Feature_en_US.html from the WebFashion directory to your store directory (for example, c:\ibm\wcs\samples\stores\itsodemo).

11. Update the Store Services sample list. This is done by editing the XML file that identifies the samples as follows:

    a. Open the file <wcs_install_path>\xml\tools\devtools\SARRegistry.xml using a text editor or an XML editor.

    b. Add the following entry directly after the `<SAR-properties>` tag:

```
<SampleSAR fileName="ITSODEMO.sar" relativePath="ITSODEMO">
<html locale="en_US" featureFile="ITSODEMO/Feature_en_US.html"
sampleSite="RetailModel/preview/en_US/index.html"/> </SampleSAR>
```

c. Save and close the file.

12. Stop and start the WebSphere Commerce Server - <wcs_instance> from the WebSphere Administrator's Console.

You have just created a template from which you can generate other stores.

11.2.2 Generate a store from the template

To create a store from the template created in the previous section, complete the following steps:

1. Start Store Services by using Internet Explorer V5.5 (or later) and entering the following URL: `http://<your_host_name>/storeservices`.
2. In the Store Services Logon page, enter your WCS administrator name and password, then click **Log On**.
3. From the Store Archive List page, click **New**.

> **Note:** The first time you access this JSP it will need to be compiled (be patient).

4. When the **Create Store Archive** page appears, enter the following data, and then click **OK**:
   - Store archive: ITSODEMO1

     This is the name of the new store archive. After you have created the store archive, the file extension .sar is added to the name of the store archive, for example, ITSODEMO1.sar.
   - Store directory: ITSODEMO1

     The directory name defines where the Web assets will be published on the server.
   - From the Store owner drop-down list, select the organization that owns the store. Select the organization you created (for example, itso).
   - From the Sample list box, select **ITSODEMO.sar**.
5. You will receive a message that ITSODEMO1.sar was created successfully. Click **OK**.

You have now created the store archive that will be used for all our following examples.

11.2.3 Publish the new store

Now that an archive exists for the store, we will publish it to the WebSphere Commerce Server runtime so that we can verify our changes as we develop.

1. Check the **ITSODEMO1.sar** box, and then click **Publish...** on the right-hand side of the window.
2. In the Publish Store Archive window, take note of the directory paths specified, then click **OK**.
3. The Store Archive List should reappear, and the status of the new store should read **Publishing**.
4. Wait for about 30 seconds then click **Refresh** on the right side of the window. Repeat this process until the status says `Publishing completed successfully.`
5. Your new store is now published and running. To see the running store, complete the following steps:
   a. Check the **ITSODEMO1.sar** box, and then click **Publish Summary** on the right-hand side of the window.
   b. At the bottom of the Publish Summary window, click the button **Launch the Store**.
   c. Your store will appear in a new browser window.

   Tip: We recommend that you bookmark this page. The URL is very long, and you will want to view your store home page often as you develop the store.

6. To enable WCS for discounts, complete the following steps:
   a. Open a DB2 command window.
   b. Navigate to the <wcs_install_path>/bin directory.
   c. Type the following at the DB2 command line:

      ```
      discount.db2 <store_name> <wcs_db_name> <db_user> <db_password>
      ```

      For example:

      ```
      discount.db2 ITSDEMO1 wcs admin admin
      ```

11.2.4 Updating VAJ WTE to run new store

In our sample we used WebFashion to create a new store called ITSODEMO. In order to run the ITSODEMO in VAJ WTE, complete the following steps:

- Copy store directory to VAJ:

  From: <wcs_install_path>\stores\web\ITSODEMO1

  To: <vaj_install_path>\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host\default_app\web

- Update the store name and servlet path for the file default_app.webapp in the following directory:

  <vaj_install_path>\ide\project_resources\IBM WebSphereTest Environment\hosts\default_host\default_app\servlets

- Create new project Web Fashion
- Import WebFashion.jar into project
- Add classpath to Servlet Engine

  a. In the WebSphere Test Environment Control Center, click **Servlet Engine**, then, in the Servers window, click **Edit Classpath**.

  b. Click **Select All** (Web Fashion).

  c. Check the **Display trace messages** box, then click **Apply**.

  d. Click **Start Servlet Engine**.

Note: If you remove any project from your workspace and put it back in later, you need to add that project back into your Servlet Engine classpath. The WebSphere Test Environment does not include new projects into the Servlet Engine classpath automatically.

11.3 Load store assets into Commerce Studio

This section provides instructions for adding store assets to a WebSphere Commerce Studio project. Using Commerce Studio, you can edit the store pages and other file assets using Page Designer, other Studio tools, or with tools registered with Studio such as XML Spy. From Studio you can publish the files to a VAJ WTE, a WCS runtime environment for testing, or to the local file system in preparation for creating your own SAR file.

Note: At the time of writing this redbook, the Commerce Studio extensions for import and export only support the Web application store assets (webapp.zip). For this reason, we unzip all SAR store assets manually to the local file system and then import these assets to Studio manually.

The procedure to load store assets into WebSphere Commerce Studio is organized as follows:

- Unzip SAR to PackageSAR directory
- Create a Studio project
- Import store assets into Studio

11.3.1 Unzip SAR to PackageSAR directory

Prior to importing store assets into Studio, we need to unzip all the assets from the SAR to a directory on the local file system, by completing the following steps:

1. Install WinZip or other zip utility.

   This will be used to manually unzip assets from the SAR file.

   If you need to download and register for the WinZip utility, it can be found at:

   http://download.cnet.com/

2. Create a directory PackageSAR to contain the SAR file:

   a. Create a folder named PackageSAR.

      This can be created anywhere that is accessible and that you will remember (for example, C:\PackageSAR).

   b. Create a folder named <your_store_name>.

      For example, C:\PackageSAR\ITSODEMO1

      This directory structure should be created on your hard drive in order to maintain all files needed to rebuild the SAR file after customization. The folders will also be used as publishing targets in WC Studio.

3. Extract the contents of your SAR as follows:

   a. Open <wcs_install_path>\stores\web\ITSODEMO1.sar using WinZip.

   b. Click **Extract.**

   c. Ensure that the **Use folder names** checkbox is checked and extract all files excepts webapp.zip and properties.zip to the PackageSAR folder. Create a properties folder within the store directory.

      For example, C:\PackageSAR\ITSODEMO1\properties directory.

   d. Extract the content of properties.zip into the new properties folder.

   e. Create a webapp folder within the store directory.

      For example, C:\PackageSAR\ITSODEMO1\webapp directory.

   f. Extract the contents of webapp.zip into the new webapp folder.

4. When complete, the PackageSAR directory structure should look like Figure 11-1.

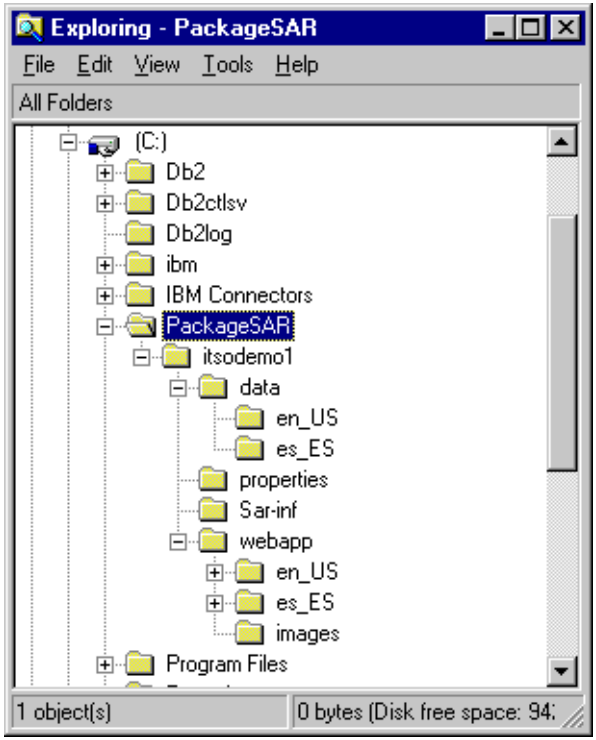


Figure 11-1 PackageSAR directory structure

11.3.2 Create a Studio project

To create a new Studio project and load the Web assets from a Store Archive, complete the following steps:

1. Start WebSphere Studio by clicking **Start** -> **Programs** -> **IBM WebSphere** -> **Studio 3.5** -> **IBM WebSphere Studio V3.5**.
2. If you are opening WebSphere Studio for the first time, or no current project exists in Studio, the Welcome to IBM WebSphere Studio window will appear. Accept the default template and then click **OK**.

Note: At the time of writing this redbook, the automated import of a SAR file does not import all of the contents of a SAR file into Studio. Certain key elements need to be imported manually. The Store Archive imports/exports only worked with Web assets (webapp.zip). It does not import the properties files (properties.zip), descriptor file (sar-inf.xml), or catalog data assets (XML and DTD import files).

To work around this problem, we have provided manual procedure for importing all store assets into Studio for development purposes.

3. When the New Project window appears, enter the following:
   - Project Name: `ITSODEMO`
   - Project Folder: `c:\files\studio\itsodemo`
   - Click **OK**

11.3.3 Import store assets into Studio

This section describes the steps necessary to manually import store assets that are not automatically imported from the Store Archive. These files have been unziped from the SAR file to the PackageSAR directory into WebSphere Commerce Studio.

The import of the store assets into Studio is organized as follows:

- Create Studio directory structure for manual import
- Import Web assets
- Import properties assets
- Import catalog data assets
- Import sarinfo.xml descriptor

Create Studio directory structure for manual import

In this section, we will create the Studio directory structure in preparation for importing the store assets as seen in Table 11-2, where itsodemo is the project name. The directory structure is consistent with the form needed for publishing the files to create a SAR and publish to WCS and WTE.

Table 11-2 Studio directory structure for manual import of assets

Store asset	Example Studio project directory structure	Description
Web assets	itsodemo\webapp	JSPs
	itsodemo\webapp\images	Images
	itsodemo\webapp\en_US\images	Locale specific images for en_US
	itsodemo\webapp\es_ES\images	Locale specific images for es_ES
Properties	itsodemo\properties	Locale specific properties files
Catalog data	itsodemo\data	Catalog data XML, DTD files
	itsodemo\data\en_US	Locale specific catalog XML files
	itsodemo\data\es_ES	Locale specific catalog XML files
SAR descriptor	itsodemo\SAR-INF	Location of sarinfo.xml
	itsodemo\SAR-INF\sar-inf	Location of sarinfo.dtd, non-publishable

To create the Studio directory structure, complete the following steps:

1. Select <your_studio> project from the left-hand pane of Studio. For example, we selected itsodemo. Right-click **Insert** -> **Folder**.
2. When the Insert Folder window appears, enter `webapp` in the Folder Name field, and then click **OK**.
3. Repeat this process to create all folders (directories) in Table 11-2, for the Studio directory structure required prior to importing store assets.

Note: The SAR-INF directory must be uppercase. All other directories are lowercase. This is cosmetic within Studio and the Windows NT filesystem and mandatory when creating the directories and zip files within the SAR.

4. When complete, your directory structure in Studio should look like Figure 11-2.

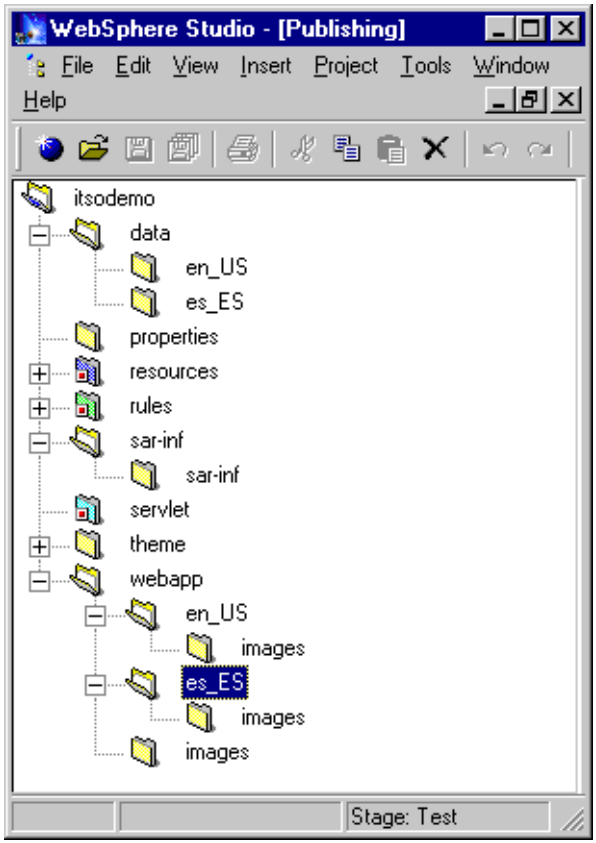


Figure 11-2 Studio directory structure

Import Web assets

To import the Web assets into Studio for modification, complete the following steps:

1. To import the Web asset JSP files:
   a. Select the **webapp** folder. Right-click **Insert** -> **File**.
   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.
   c. From the Open window, navigate to the C:\PackageSAR\<your_store>\webapp directory. Press Ctrl+A to select all files, and then click **Open**.

Studio will not copy folders.

d. When you return to the Insert File window, all the files should be listed in the Files field. Click **OK** to import the files.

2. To import the Web asset image files:

   a. Select the **images** folder under webapp. Right-click **Insert** -> **File**.

   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.

   c. From the Open window, navigate to the `C:\PackageSAR\<your_store>\webapp\images` directory. Press **Ctrl+A** to select all files, and then click **Open**.

   d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.

3. To import the Web asset locale specific image files:

   a. Select the **images** folder under webapp\<locale>. Right-click **Insert** -> **File**. For example we selected the \webapp\en_US\images folder.

   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.

   c. From the Open window, navigate to the `C:\PackageSAR\<your_store>\webapp\en_US\images` directory. Press Ctrl+A to select all files, and then click **Open**.

   d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.

   e. Repeat this process for additional locale specific images (es_ES).

Import properties assets

To import the properties file assets into Studio for modification, complete the following steps:

1. Select the **Properties** folder. Right-click **Insert** -> **File**.
2. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.
3. From the Open window, navigate to the C:\PackageSAR\<your_store>\properties directory. Press Ctrl+A to select all files, and then click **Open**.
4. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.

Import catalog data assets

To import the catalog data file assets into Studio for modification, complete the following steps:

1. To import the catalog data XML files:
   a. Select the **data** folder. Right-click **Insert** -> **File**.
   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.
   c. From the Open window, navigate to the C:\PackageSAR\<your_store>\data directory. Press Ctrl+A to select all files, and then click **Open**.
   d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.
2. To import the catalog data XML file corresponding DTD files:
   a. Select the **data** folder. Right-click **Insert** -> **File**.
   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.
   c. From the Open window, navigate to the C:\<wcs_install_path>\xml\sar directory. Press Ctrl+A to select all DTD files, and then click **Open**.
   d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.
3. To import the locale specific catalog data XML files:
   a. Select the **en_US** folder within the data folder. Right-click **Insert** -> **File**.
   b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.
   c. From the Open window, navigate to the C:\PackageSAR\<your_store>\data\<locale> directory (for example, en_US). Press Ctrl+A to select all files, and then click **Open**.
   d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.
   e. Repeat this process to add assets for additional locales (es_ES).

Import sarinfo.xml descriptor

To import the sarinfo.xml and corresponding DTD file into Studio, complete the following steps:

1. To import the sarinfo.xml file:
   a. Select the **SAR-INF** folder. Right-click **Insert** -> **File**.

b. When the Insert File window appears, click the **Use Existing** tab, and then click **Browse**.

c. From the Open window, navigate to the C:\PackageSAR\<your_store>\SAR-INF directory. Select the sarinfo.xml file, and then click **Open**.

d. When you return to the Insert File window all the files should be listed in the Files field. Click **OK** to import the files.

2. To import the sarinfo.dtd file:

a. Drag and drop the sarinfo.dtd file from the C:\<wcs_install_path>\xml\sar folder to the SAR-INF folder within the SAR-INF folder.

b. Select the **sarinfo.dtd** file in the SAR-INF folder. Right-click and select **Properties**. Select the **Publishing** tab. Ensure the **Set publishable** checkbox is not checked.

11.3.4 Studio configuration after import of assets

After you have imported the store assets, we need to configure Studio. This section includes the following sub sections:

- Tools registration of XML editor
- Associating DTD files with XML files
- Turn off syntax checking in Page Designer
- Set publishing preferences

Tools registration of XML editor

If you installed XML Spy and want to edit files directly from Studio, you can register XML Spy to be the default editor for XML files, by completing the following steps:

1. Start WebSphere Studio.
2. Click **Tools** -> **Tools Registration** from the menu bar.
3. When the Tools Registration window appears, enter `xml` in the file extension file, and then click **Edit**. Notice that Notepad is the default editor.
4. When the Edit object type window appears, select the **Editing Applications** tab.

   XML Spy does not appear in the Registered applications list.
5. Click **Browse** to find the XML Spy executable.
6. Select **XMLspy.exe** from the directory that you installed XML Spy, and then click **Open**.

7. From the Editors pane, select **XMLspy.exe** and then click **Set as Default**.
8. Click **OK**.
9. Repeat this process for other file extensions and tools.

Associating DTD files with XML files

For each XML file you edit, there will most likely be a DTD file of the same file name. In addition, when you open the XML file with the XML Spy or other XML, you may see a message stating `the <filename>.dtd could not be opened`.

To associate a DTD file with an XML file, complete the following steps:

1. Open the XML file of interest, take note of DTD files that could not be opened.
2. Select the DTD file that could not be opened. Right-click and select **Insert** -> **Source Link**.
3. Check the XML file to associate as Parent and click **OK**.
4. Repeat this process for other DTD files needed by the XML file.

Turn off syntax checking in Page Designer

The store pages in the WebFashion sample store use non-standard syntax. Page Designer cannot reconcile URL commands. For this reason, we will turn off syntax checking in WebSphere Studio Page Designer.

To turn off the syntax checking in WebSphere Studio Page Designer, complete the following steps:

1. From the left-hand pane, select and expand **itsodemo** -> **stores** -> **ITSODEMO1**. Double-click one of the JSP files (for example, account.jsp). The non-standard syntax error message displays.
2. From the Tools menu, select **Page Designer Options**.
3. When the Options window appears, select the **General** tab.
   a. Ensure that the Format HTML source automatically and Correct HTML syntax errors automatically fields are *not selected*.
   b. From the drop-down list, select **Ignore all errors**.
   c. Click **Apply**.
   d. Click **OK**.
4. Close the JSP file.

Set publishing preferences

This step needs to be done only once per Studio project. It can, however, be repeated if your publishing preferences change. The following example details how to set the project to update the store archive when Web assets are changed.

1. In WebSphere Studio, select the **ITSODEMO** project (the highest level folder).
2. From the Tools menu, select **Wizards**, then **Store Archive Publish Preferences**. The Store Archive Publish Preferences window opens as shown below in Figure 11-3.

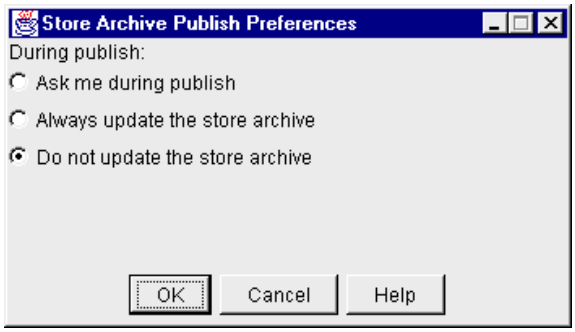


Figure 11-3 Store Archive Publish Preferences

3. Select **Do not update the store archive**. We will recreate the store archive ourselves in Chapter 12, “Creating a store archive (SAR) for deployment” on page 355.
4. Click **OK**.

11.4 Configuring Studio for publishing

Most of the changes for which we provide examples in this chapter are cosmetic changes. For the most part, they involve only property files, images, HTML and JSP changes.

Attention: Be *warned* that during re-publishing your store using Store Serivces, command line or via Studio (exception: publishing single files in Studio!) your assets in the following folders will be deleted automatically, because WCS deletes the whole folders including all content and recreates them new.

- Your store's property folder under <wcs>\stores\properties\
- Your store's folder and subfolders under <wcs>\stores\web\

The publishing of assets from Studio is organized into the following sections:

- Create publishing stage
- Create a server to publish to
- SAR: define assets to publish and target publishing path
- WCS: define assets to publish and target publishing path
- WTE: define assets to publish and target publishing path

11.4.1 Create publishing stage

In this section we will create publishing stages for the store assets for the WCS runtime, WTE, and PackageSAR directory as seen in Table 11-3.

Table 11-3 Stage and server for publishing

Stage name	Server name	Description
WCS	<hostname>_WCS	Publish the Web assets to a running store for testing purposes.
WTE	<hostname>_WTE	Publish the Web assets to the VAJ WTE for test and debug purposes.
SAR	<hostname>_SAR	In this case, all store assets (Web assets, SAR descriptor, data files, properties files) are published to the the PackageSAR directory on the local file system in preparation for building a deployable SAR file.

To create publishing stages, complete the following steps:

1. Select **Project** -> **Customize Publishing Stages** from the Studio menu bar.
2. When the Customize Publishing Stages window appears, enter the stage names listed in Table 11-3. For example, enter `SAR` in the Stage name field and then click **Add**.

3. Repeat this process to add all the stage names (WCS, WTE, SAR).
4. When done adding all the publishing stages, click **OK** to close window.

11.4.2 Create a server to publish to

To create a server that you can publish the store assets to within each of the stages created in the previous section, complete the following steps:

1. Select **Project** -> **Publishing Stage** -> **<stage_name>** from the Studio menu bar to select the current stage. For example, we selected stage name SAR.
2. From the right-hand pane, select the stage name **SAR**.
3. Right-click **Insert** -> **Server**.
4. When the Insert Server window appears, enter the following:
   - Server name: `<hostname>_<stage_name>`

     This is completely arbitrary. For example, we entered `m23vnx58_SAR`.
   - Server address:

     Enter the hostname of your target server for this stage. For example, we entered `http://m23vnx58`.

   Click **OK**.
5. Repeat this process to add server to the other publishing stages described in Table 11-3 on page 339.

11.4.3 SAR: define assets to publish and target publishing path

Depending on the stage and server to publish to, you will need to define different assets to publish. By default the first server added to a stage will be the default server and list all folders in the project (left-hand pane). We will need to delete and add folders for each stage and server.

To define the store assets to publish and target path to publish the assets for the SAR stage and <hostname>_SAR server, complete the following steps.

Define assets to publish (SAR)

1. Select the **SAR** stage by clicking **Project** -> **Publishing Stage**.
2. Delete the folders named “theme” and “stores”.
   a. From the right-hand pane, under <your_hostname>_SAR server, select the **theme** folder. Right-click **Delete**.
   b. When the Delete window appears, select **Delete from current stage** and click **OK**.

3. The assets to publish for packaging a SAR should look like Figure 11-4.

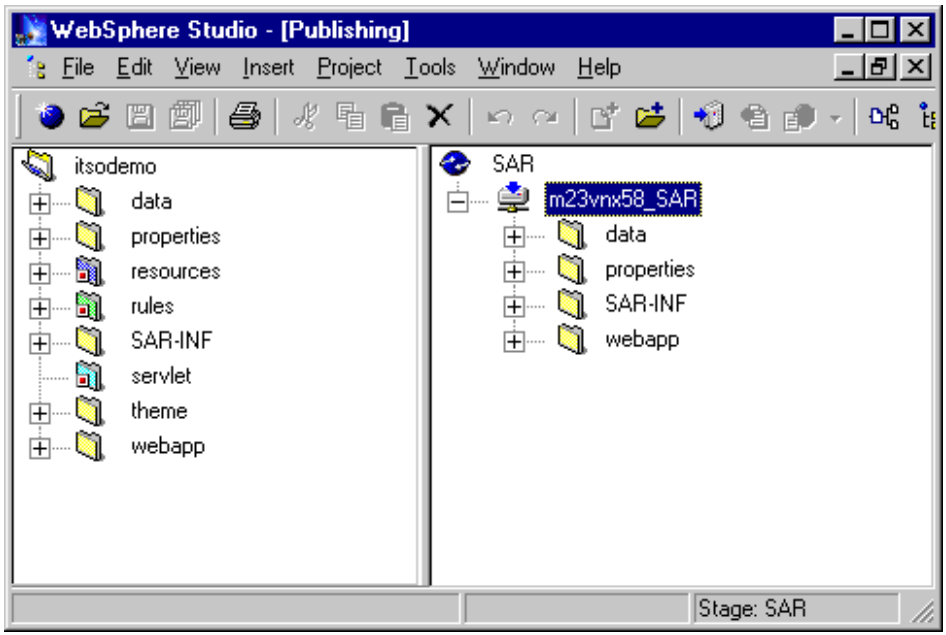


Figure 11-4 SAR assets to publish

Define target publishing path (SAR)

1. Select the **<hostname>_SAR** server (for example, m23vnx58_SAR).
2. Right-click and select **Properties**.
3. When the Properties window appears, click **Targets**.
4. When the Publishing Targets window appears, click **Add**. Enter the following:
   - Name: `PackageSARroot`
   - Path: `c:\PackageSAR\<your_store>`

     For example, we entered c:\PackageSAR\itsodemo1.

Note: All assets defined to be published will be published to the PackageSARroot, unless otherwise specified.

5. Click **OK** to save settings and return to Properties window. From the Publishing target pull-down, select **PackageSARroot**.
6. Click **OK** to close the Properties window.

7. For each folder (data, properties, SAR-INF, webapp) under the <hostname>_SAR server, do the following:
   a. Select the folder (for example, data). Right-click **Properties**.
   b. When the Properties window appears, do the following:
      - Make sure Publish this folder to a publishing target is *not* checked.
      - Make sure this folder a virtual directory is *not* checked.
      - Click **OK**.
   c. Repeat this process for all the folders (data, properties, SAR-INF, webapp).

11.4.4 WCS: define assets to publish and target publishing path

When you are creating or modifying JSPs within Studio, we use Studio to publish the modified JSPs directly to the WCS runtime (no need to package SAR) for testing purposes.

To define the store assets to publish and target path to publish the assets for the WCS stage and <hostname>_WCS server, complete the following steps:

Define assets to publish (WCS)

1. Select the **WCS** stage by clicking **Project** -> **Publishing Stage**.
2. Delete folders (theme, data, stores, SAR-INF).
   a. From the right-hand pane, under <your_hostname>_WCS server, select the **theme** folder. Right-click **Delete**.
   b. When the Delete window appears, select **Delete from current stage** and click **OK**.
   c. Repeat process to delete the data and SAR-INF folders from the right-hand pane.

      These are not needed on the file system of the WCS runtime. They are only when populating the WCS database.
3. Create folder webapp.
   a. Drag and drop <your_store> (for example, ITSODEMO1) from the right-hand pane under stores to the <your_hostname>_WCS server in the right-hand pane.
   b. Select the **ITSODEMO1** folder in the right-hand pane. Right-click and select **Rename**. Rename the folder to `webapp`.
4. When done, you should have a webapp and properties folder.

Define publishing target (WCS)

1. Select the **<hostname>_WCS** server (for example, m23vnx58_WCS).
2. Right-click and select **Properties**.
3. When the Properties window appears, click **Targets**.

   The Publishing Targets window should appear.
4. To define the WCS Web assets target path, click **Add** and enter the following:
   - Name: `WCSWeb`
   - Path: `c:\<wcs_install_path>\stores\web\<your_store>`

     For example, we entered `c:\ibm\wcs\stores\web\itsodemo1`.
   - Click **OK**.
5. To define the WCS properties files target path, click **Add** and enter the following:
   - Name: WCS`Properties`
   - Path: `c:\<wcs_install_path>\stores\properties\<your_store>`

     For example, we entered `c:\ibm\wcs\stores\properties\itsodemo1`.
   - Click **OK**.
6. Click **OK** to save settings. Click **OK** to close the Properties window.
7. To define publishing target for the contents of the webapp folder.
   a. Select the **webapp** folder from the right-hand pane. Right-click **Properties**.
   b. When the Properties window appears, do the following:
      - Select **Publish this folder to a publishing target**
      - Select **WCSWeb** from the pull-down
      - Check **Make sure this folder a virtual directory**

        In this case, we do not want the folder name webapp.
      - Click **OK**.
8. To define publishing target for the contents of the properties folder.
   a. Select the **properties** folder from the right-hand pane. Right-click **Properties**.
   b. When the Properties window appears, do the following:
      - Select **Publish this folder to a publishing target**
      - Select **WCSProperties** from the pull-down
      - Check **Make sure this folder a virtual directory**

In this case, we do not want the folder name properties.

- Click **OK**.

11.4.5 WTE: define assets to publish and target publishing path

When you are creating or modifying JSPs within Studio, and want to publish files the VAJ WTE environment, we use Studio to publish the modified JSPs for debug and testing purposes. This setup is nearly identical to the WCS setup, with the exception of the publishing target path.

To define the store assets to publish and target path to publish the assets for the WTE stage and <hostname>_WTE server, complete the following steps.

Define assets to publish (WTE)

1. Select the **WCS** stage by clicking **Project** -> **Publishing Stage**.
2. Delete folders (theme, data, stores, SAR-INF).
   a. From the right-hand pane, under <your_hostname>_WTE server, select the **theme** folder. Right-click **Delete**.
   b. When the Delete window appears, select **Delete from current stage** and click **OK**.
   c. Repeat process to delete the data and SAR-INF folders from the right-hand pane.
3. Create a folder called “webapp”.
   a. Drag and drop <your_store> (for example, ITSODEMO1) from the right-hand pane under stores to the <your_hostname>_WTE server in the right-hand pane.
   b. Select the **ITSODEMO1** folder in the right-hand pane. Right-click and select **Rename**. Rename the folder to webapp.
4. When done, you should have a webapp and properties folder.

Define publishing target (WTE)

1. Select the **<hostname>_WTE** server (for example, m23vnx58_WTE).
2. Right-click and select **Properties**.
3. When the Properties window appears, click **Targets**.
4. To define the WTE Web assets target path, click **Add** and enter the following:
   – Name: `WTEWeb`
   – Path: `<vaj_install_path>\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host \default_app\web \<store_name>`

For example, we entered `c:\ibm\vaj\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host \default_app\web \itsodemo1`

– Click **OK**.

5. To define the WTE properties files target path, click **Add** and enter the following:
   – Name: `WTEProperties`
   – Path: `<vaj_install_path>\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host \default_app\web \<store_name>`

     For example, we entered `c:\ibm\vaj\ide\project_resources\IBM WebSphere Test Environment\hosts\default_host \default_app\web \itsodemo1`

   – Click **OK**.
6. Click **OK** to save settings. Click **OK** to close the Properties window.
7. To define publishing target for the contents of the webapp folder.
   a. Select the **webapp** folder from the right-hand pane. Right-click **Properties**.
   b. When the Properties window appears, do the following:
      • Select **Publish this folder to a publishing target**
      • Select **WTEWeb** from the pull-down
      • Check **Make sure this folder a virtual directory**

        In this case, we do not want the folder name properties.
      • Click **OK**.
8. To define publishing target for the contents of the properties folder.
   a. Select the **properties** folder from the right-hand pane. Right-click **Properties**.
   b. When the Properties window appears, do the following:
      • Select **Publish this folder to a publishing target**
      • Select **WTEProperties** from the pull-down
      • Check **Make sure this folder a virtual directory**

        In this case, we do not want the folder name properties.
      • Click **OK**.

11.5 Publishing from Studio

WebSphere Studio provides the capability to publish one, several, or all assets in a project. In addition, by setting up different stages and servers within Studio, as we have done, you can publish different store assets to different servers and target paths. In our example, we publish to the WCS runtime, WTE, and the PackageSAR directory on the local file system.

11.5.1 Publish all assets defined for a stage and server

To publish all the store assets for the selected stage and server, complete the following steps:

1. Ensure that all files are *checked in*.
   - Files that are checked out of Studio (for example, files that are in use, or have been opened) are marked with a red checkmark.
   - To check in a file, right-click it and from the drop-down menu select **Check In**.
2. Select the desired stage for publishing. For example, if we want to publish to the WCS runtime, select the WCS stage.
3. From the right-hand pane, select <hostname>_WCS (server for stage).
4. Right-click and select **Publish this Server.**
5. When the Publishing Options window appears, you may want to change the following options:
   - Deselect **Publish only modified files**.
   - Click **OK**.

   Changing these settings is optional.
6. When the Files to Publish window appears, review the list. All files are checked by default. Click **OK** to publish files.
7. You may get a message that the files already exist, depending on the location of your publishing target.
8. When the Publish Report window appears, review the report for any problems encountered during publishing.

11.5.2 Publish selected files

If you are modifying a JSP and want to test it, it does not make sense to create a new SAR file to test one file. You can publish just the one file to the WCS runtime or the WTE for testing purposes.

To publish selected files from Studio, complete the following steps:

1. Select the file or files of interest.
2. Right-click and select **Publish this File**.
3. The Publishing Options window will appear, accept the default and click **OK**.
4. When the Files to Publish window appears, review the list. All files are checked by default. Click **OK** to publish files.
5. You may get a message that the files already exist, depending on the location of your publishing target.
6. When the Publish Report window appears, review the report for any problems encountered during publishing.

11.5.3 Creating a SAR

Our WebSphere Studio environment is set up so that the contents and directory structure of a SAR file can be readily re-created by publishing to the PackageSAR directory.

In Chapter 12, “Creating a store archive (SAR) for deployment” on page 355 we explain how to create a SAR from the published output of Studio.

11.6 Basic customization of Web assets

This section outlines the procedure to follow when customizing Web assets. We have included the following basic examples:

- Example 1: changing the store logo
- Example 2: changing a label
- Example 3: changing a properties files
- Example 4: changing the descriptor file

Note: This redbook covers only *basic* customization using WebSphere Commerce Studio V5.1, Professional Developer Edition for Windows NT and Windows 2000. Our intention is to document the procedures for modifying the various assets with the WebSphere Commerce Studio tools.

- Store Services customizations

  For examples on customizing store profile information, shipping information, and taxes using Store Services refer to WCS online help or download the tutorial *Creating an online store*, updated on 21 February 2001, from:

  http://www-4.ibm.com/software/webservers/commerce/wcs_pro/lit-tech-general.html

- Advanced customizations using Commerce Studio

  For more advanced customizations, refer to the *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174.

11.6.1 Example 1: changing the store logo

To change the store logo, select a new logo for your store and edit the header.jsp file:

1. Select a new logo for your store.

   For example, using the Windows Explorer, choose the new logo <studio_install_path>\samples\image\illust\b_cut001.gif.

2. In the left pane in WC Studio, expand **stores** -> **ITSODEMO1** -> **en_US**.
3. Right-click **images** and select **Insert** -> **File**.
4. When the Insert File window appears, select the **Use Existing** tab. Click **Browse**, and navigate to the file you selected in step 1.
5. Click **Open**, and then **OK**.
6. When the new logo file opens in the images directory, in the left pane of Studio expand **stores** -> **ITSODEMO1**.
7. Double-click **header.jsp** so that it opens in Page Designer in the right pane.
8. Select the **HTML Source** tab.
9. Find the image tag as follows:

```
<img src="<%=storeDir%>/<%=locale.toString()%>/images/logo.gif"
alt="FashionFair" width="30" height="30" align="center">
```

10. Replace logo.gif with the name of your new logo file.
11. Close header.jsp, saving your changes.

12. Locate header.jsp in the left pane of the window. A red checkmark besides it indicates that the file is still checked out. To check it in, right-click and then select **Check-in**.
13. In order to see the new store logo, you must publish the changed header.jsp, and the new logo.

    Refer to 11.4, “Configuring Studio for publishing” on page 338 for details on publishing.
14. Open a browser and enter the URL for the store’s home page to see the new logo.

11.6.2 Example 2: changing a label

In the following example, we change a label in the file myaccount.jsp.

Note: This example belongs together with the example on how to change and create a new label, performed in 11.6.3, “Example 3: changing a properties files” on page 350.

You won’t be able to review the changes made in *this* section unless you performed the changes detailed in 11.6.3, “Example 3: changing a properties files” on page 350.

1. In the left pane of Studio, expand **stores** -> **ITSODEMO1**.
2. Double-click **myaccount.jsp** to open it in Page Designer, visible in the right-hand pane.
3. Select the **HTML Source** tab.
4. Find the label for logonId as follows:

   ```
   <%=infashiontext.getString("EMAIL_ADDRESS2")%>
   ```
5. Replace EMAIL_ADDRESS2 with LOGONID3. It should look like this now:

   ```
   <%=infashiontext.getString("LOGONID3")%>
   ```
6. Close myaccount.jsp save your changes.
7. Locate myaccount.jsp in the left-hand pane. A red checkmark besides it indicates, that the file is still checked out. To check it in, right-click and then select **Check-in**.
8. In order to see the new store logo, you must publish the changed file myaccount.jsp.

   Refer to 11.4, “Configuring Studio for publishing” on page 338 for details on publishing.

9. Open a browser and enter the URL for the store's home page and then click on **MY ACCOUNT** to see the effects of your change.

11.6.3 Example 3: changing a properties files

This section will guide you through the process of changing a property file. As property files mainly contain textual information to be assembled at runtime by the JSP, they are extremely important for multicultural enablement in WCS.

In preparation for the two working examples to follow in Chapter 17, "SecureWay Directory (LDAP)" on page 471, we will create a new label for an entry field used in the Web Fashion store registration procedure.

Note: This example belongs together with the example on how to change an appropriate JSP to use a different label, performed in 11.6, "Basic customization of Web assets" on page 347.

You won't be able to review the changes made in *this* section unless you performed the above-mentioned example.

To create a new field in a property file, complete the following high-level steps:

1. In this example we change a label that appears on the **Register or Login** of the Web Fashion store. To view the page before the change, open the Web Fashion store in your browser and click the **Register link** in the left-hand frame. The Register or Login page displays. Look for the entry field called E-mail address.
2. Open WebSphere Studio by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Studio** -> **Studio 3.5** -> **IBM WebSphere Studio V3.5**.
3. Expand the **stores** -> **ITSODEMO1** from the left-hand pane.
4. Expand the **Properties** tree and double-click on **infashiontext_en_US.properties.**
5. Edit the line `EMAIL_ADDRESS2 = E-mail address:` below the line `#myaccount.jsp`.
   - Change the entry so that it reads:

     ```
     EMAIL_ADDRESS2 = My e-mail address:
     ```

   - Insert a line beneath as follows:

     ```
     LOGONID3 = My desired LogonID:
     ```

   - Save and close the Properties file.
6. Right-click **infashiontext_en_US.properties** and select **Check In**.

You have successfully changed a property file and also created a new label.

7. In order to view the modified `EMAIL_ADDRESS2` label, you must publish the changed file, **infashiontext_en_US.properties**.

   Refer to 11.4, “Configuring Studio for publishing” on page 338 for detailed instructions on publishing.

8. Open a browser and enter the store’s home page URL, then click on **MY ACCOUNT** to see the effects of your change.

Note: In order to view the page with the newly-created label, complete the steps detailed in 11.6.2, “Example 2: changing a label” on page 349.

We experienced that restarting the WebSphere Commerce Server - <instance> application server is mandatory for the changes in the properties files to be in effect.

11.6.4 Example 4: changing the descriptor file

Each store archive must include a sarinfo.xml file. This file is known as the *descriptor*. The descriptor contains information about the store archive that is used when a store archive is published, including the names of the file asset ZIP files and the store database XML files, and the order in which they are published. If a store archive includes files in multiple languages, the sarinfo.xml file also includes that information and determines the order in which each language file is published.

To change the sarinfo.xml file, complete the following steps:

1. In the publishing view, select and expand the **SAR-INF** folder.
2. Modify the sarinfo.xml.
   a. Double-click **sarinfo.xml** file to open it in a text editor, or if you registered your XML editor with Studio, the file will open with the XML editor.
   b. For this example we want to disable the fragmentation of campaigns used in our demo store. This means WCS will use the same settings for all available languages. Find the piece of code in Example 11-1 in the sarinfo.xml file.

Example 11-1 Sample campaign for demo store

```
<asset
       fragmented="yes"
       name="campaign">
       <file
          name="data/campaign.dtd"
          type="dtd"/>
```

```
    <file
       name="data/campaign.xml"
       priority="20"
       type="db-load"/>
    <file
       name="data/en_US/campaign.xml"
       priority="24"
       type="db-load">
       <locale>en_US</locale>
    </file>
    <file
       name="data/es_ES/campaign.xml"
       priority="24"
       type="db-load">
       <locale>es_ES</locale>
    </file>
 </asset>
```

c. Simply change `fragmented="yes"` to `fragmented="no".`

d. Save your changes and perform a file check-in.

3. You have successfully changed the descriptor file.

For more information on the elements, attributes and attribute values, please refer to the WCS online help.

Note: For more information on the XML specifications for a store archive, examine the sarinfo.dtd in `<WCS_install_path>\xml\sar` with your XML editor and refer to WCS online help.

11.7 Where to find more information

- WCS online help
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174
- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*
- *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*

- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*
- WCS homepage tech library

  http://www-4.ibm.com/software/webservers/commerce/wcs_pro/lit-tech-general.html
- WCS homepage library

  http://www.ibm.com/software/webservers/commerce/wcs_pro/lit.html
- For more information and case studies about B2C visit IBM's B2C e-commerce solutions Web page on:

  http://www.ibm.com/software/webservers/commerce/btoc/
- For more information and case studies about B2B visit IBM's B2B e-commerce solutions Web page on:

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

12

Creating a store archive (SAR) for deployment

There are many ways that someone could create or re-create a SAR file for publishing. In this chapter we provide a methodology and a working example for creating a SAR from a directory tree created by publishing the store assets from WebSphere Studio to the local file system, and then packaging the files into a SAR file using a zip utility.

This chapter is organized into the following sections:

- Overview
- Publish Studio project
- Packaging a SAR
- Deploying a SAR
- Verify store
- Where to find more information

© Copyright IBM Corp. 2001

12.1 Overview

In all of the Studio examples, we have chosen not to export assets back into the SAR file. The reason for this is that only Web assets can be easily transferred this way. Any additional assets require manual attention. As such, we have chosen to complete the entire process manually.

We have outlined the process for creating a SAR as follows:

1. Publish Studio project
   a. Create PackageSAR directory
   b. Importing or creating store assets in Studio
   c. Create SAR packaging stage
   d. Create server for SAR packaging stage
   e. Defining assets to publish and publishing targets
   f. Publish Studio project to PackageSAR directory
2. Packaging a SAR
   a. Packaging a SAR from the command line using PKZIP

   or

   b. Packaging a SAR from a GUI zip utility
3. Deploying a SAR
   a. Deploy a SAR from Store Services

   or

   b. Deploy a SAR from command line
4. Verify store
   a. Clear cache
   b. Testing your store

12.2 Publish Studio project

The purpose of this section is to highlight the steps required to configure Studio for publishing the store assets to the file system in preparation for publishing.

This section is organized as follows:

- Create PackageSAR directory
- Importing or creating store assets in Studio

- Create SAR packaging stage
- Create server for SAR packaging stage
- Defining assets to publish and publishing targets

Note: If you created a store following the procedure documented in Chapter 11, “Create and customize a store using Commerce Studio” on page 321, the following steps have already been completed.

12.2.1 Create PackageSAR directory

The creation of the PackageSAR directory should have been completed as part of importing the store assets into Studio. If not, or if you are creating your store from scratch, you need to create a directory called PackageSAR. This directory will be used as the root directory to publish the store assets from Studio.

For example, we created the directory c:\PackageSAR.

Refer to 11.3.1, “Unzip SAR to PackageSAR directory” on page 329 for more detailed information.

12.2.2 Importing or creating store assets in Studio

When importing or creating assets in Studio, we need to consider the directory structure or folders within Studio. It is possible to have a folder structure on the left-hand side of Studio under the project view that is different from how you define the folder structure for publishing. However, for simplicity we recommend that you make them the same.

For example, we imported all the store assets into the directory structure (folders) within Studio as seen in Table 12-1.

Table 12-1 Studio directory structure for manual import of assets

Store asset	Example Studio project directory structure	Description
Web assets	itsodemo\webapp	JSPs
	itsodemo\webapp\images	Images
	itsodemo\webapp\en_US\images	Locale specific images for en_US
	itsodemo\webapp\es_ES\images	Locale specific images for es_ES

Store asset	Example Studio project directory structure	Description
Properties	itsodemo\properties	Locale specific properties files.
Catalog data	itsodemo\data	Catalog data xml, dtd files
	itsodemo\data\en_US	Locale specific catalog xml files
	itsodemo\data\es_ES	Locale specific catalog xml files
SAR descriptor	itsodemo\sar-inf	Location of sarinfo.xml.
	itsodemo\sar-inf\sar-inf	Location of sarinfo.dtd, non-publishable.

Refer to 11.3.3, “Import store assets into Studio” on page 331 for more detailed information.

12.2.3 Create SAR packaging stage

Within WebSphere Studio we need to define a publishing stage. For example, we created a stage called SAR. By having separate publishing stages, we can publish to different servers and file system targets without rearranging our assets within Studio.

Refer to 11.4.1, “Create publishing stage” on page 339 for more detailed information.

12.2.4 Create server for SAR packaging stage

Now that we have the SAR publishing stage, we need to insert a server to publish to. In our example, our development system included a runtime environment on the same system.

For example:

- Server name: `m23vnx58_SAR`
- Server address: `http://m23vnx58`

Note: When entering the server name, the server address is filled in automatically. If your server name is different than the hostname, you will need to update the server address field.

Refer to 11.4.2, “Create a server to publish to” on page 340 for more detailed information.

12.2.5 Defining assets to publish and publishing targets

To publish the store assets to the proper directory tree, we need to define the publishing target. In addition, when creating folders within Studio or the selection of the folders to publishing Studio can also determine where the files will be published.

For example, the following folders should be listed under the server (m23vnx58_SAR):

- data
- properties
- sar-if
- webapp

For example, we created a publishing target in the PackageSAR directory for our sample store ITSODEMO as follows:

- Publishing target name: PackageSARRoot
- Publishing target path: c:\PackageSAR\itsodemo

Refer to 11.4.3, “SAR: define assets to publish and target publishing path” on page 340 for more detailed information.

12.2.6 Publish Studio project to PackageSAR directory

Once Studio is configured for publishing, the store assets in the Studio project can be published to a directory structure on the hard disk that is consistent with the directory structure required by a SAR.

Refer to 11.5.1, “Publish all assets defined for a stage and server” on page 346 for more detailed information.

At this point you should have all the store assets published to the PackageSAR directory structure for your store.

12.3 Packaging a SAR

In this section, we provide instructions for two different methods of packaging a SAR file:

- Packaging a SAR from the command line using PKZIP

  Once the batch files are configured, a batch file is run to package the SAR. This method provides for automation and is very useful when needing to create a SAR file many times, such as during the development process.

- Packaging a SAR from a GUI zip utility

  This method offers the ability to use the Winzip GUI; however; it is manual and has a greater possibility for error.

The procedure for packaging the SAR using either method includes the following high-level steps:

1. Create webapp.zip
2. Create properties.zip
3. Create a SAR file containing the remaining assets, including the directory structure of the remaining assets, and add the webapp.zip and properties.zip.
4. When complete you should have a deployable SAR file that can be copied to the <wcs_install_path>\stores\web directory to be deployed from Store Services.

Tip: We have listed some tips for packaging a SAR file:

- For problem determination when publishing a SAR, view the <wcs_install_path>\instances\<wcs_instance>\log\ecmsg_<host>.log file.
- The \SAR-INF directory for the sarinfo.xml file must be capitalized when creating the SAR.
- The archive attribute must be set for files packaged in the SAR.

12.3.1 Packaging a SAR from the command line using PKZIP

This section describes how to package a SAR file from a command line using pkzip and batch files to automate the process.

When packaging the SAR, take note of the following points:

- Set the archive attribute
- Map the exact folder info when adding files to a SAR as described in the sarinfo.xml file (otherwise modification of this file is required).

To package a SAR file from command line, complete the following steps:

1. Install the PKZIP 4.0 Command Line utility, referred to as pkzipc.exe. Ensure that pkzipc.exe is added to the path environment variable or copied to the

PackageSAR directory where the batch files will be executed. PKZIP 4.0 Command Line is available for download at:

`http://www.pkware.com`

Note: If you want to use your own archiving utility, make sure it can handle at long filenames and extensions (such as InFashion_text_en_US.properties). Also, set the archive bit and specify the paths during the zip process.

2. Configure the batch files for packaging SAR to automate the packaging process.

   A description of these batch files as well as the target location of the batch file can be found in Table 12-2. A reference to the example contents for each batch file is included in Table 12-2 in the Name of batch file column.

Table 12-2 Batch files for SAR packaging

Name of batch file	Target path	Assets and/or action
pack_webapp.bat * see Example 12-1 for contents	\PackageSAR\itsodemo1	Compress to webapp.zip and add the zip to SAR
pack_properties.bat * see Example 12-2 for contents	\PackageSAR\itsodemo1	Compress to properties.zip and add the zip to SAR
pack_sarinfo_xml.bat * see Example 12-3 for contents	\PackageSAR	Adds the file to SAR
pack_default_xml_and_dtd.bat * see Example 12-4 for contents	\PackageSAR	Adds the files to SAR
pack_locale_en_US_xml_and_dtd.bat * see Example 12-5 for contents	\PackageSAR	Adds the files to SAR
pack_locale_es_ES_xml_and_dtd.bat * see Example 12-6 for contents	\PackageSAR	Adds the files to SAR
pack_sar.bat * see Example 12-7 for contents	\PackageSAR	Executes the other batch files

Example 12-1 pack_webapp.bat

```
echo on
rem
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem compress all web asset files to webapp.zip without path info
cd webapp
pkzipc -add -recurse -attr=archive -path=specify ..\webapp.zip  .\*.*
cd ..
rem add webapp.zip to SAR file in root of PackageSAR
pkzipc -add -nozipextension -attr=archive ..\ITSODEMO1.sar webapp.zip
```

Example 12-2 pack_properties.bat

```
echo on
rem
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem compress all properties files to properties.zip without path info
pkzipc -add -attr=archive -path=none properties.zip  .\properties\*.properties

rem add properties.zip to SAR file in the root of PackageSAR
pkzipc -add -nozipextension -attr=archive ..\ITSODEMO1.sar properties.zip
```

Example 12-3 pack_sarinfo_xml.bat

```
echo on
rem
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem add sarinfo.xml files to SAR WITH path info
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\SAR-INF\*.xml
```

Example 12-4 pack_default_xml_and_dtd.bat

```
echo on
rem
```

```
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem add all default xml and dtd files to SAR WITH path info
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\*.xml
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\*.dtd
```

Example 12-5 pack_locale_en_US_xml_and_dtd.bat

```
echo on
rem
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem add all default xml and dtd files to SAR WITH path info
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\en_US\*.xml
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\en_US\*.dtd
```

Example 12-6 pack_locale_es_ES_xml_and_dtd.bat

```
echo on
rem
rem ****************************************************************
rem * This bat has been tested to work properly with               *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.        *
rem ****************************************************************
rem
rem add all default xml and dtd files to SAR WITH path info
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\es_ES\*.xml
pkzipc -add -path=specify -nozipextension -attr=archive ..\ITSODEMO1.sar
.\data\es_ES\*.dtd
```

Example 12-7 pack_sar.bat

```
cls
echo on
rem
rem ****************************************************************
```

```
rem * This bat has been tested to work properly with                *
rem * PKZIP(R)  Version 4.00  FAST! Compression Utility for Windows *
rem * Copyright 1989-2000 PKWARE Inc.  All Rights Reserved.         *
rem ****************************************************************
rem
call pack_webapp.bat
call pack_properties.bat
call pack_locale_en_US_xml_and_dtd.bat
call pack_locale_ja_JP_xml_and_dtd.bat
call pack_locale_es_ES_xml_and_dtd.bat
call pack_default_xml_and_dtd.bat
call pack_sarinfo_xml.bat
```

3. Copy the batch files to the <install_path>\PackageSAR\<store> directory.

   For example, we copied the batch files to the c:\PackageSAR\itsodemo1 directory.

4. Modify the batch files to update the proper name for the SAR file.

   For example, the batch files are configured to create the ITSODEMO1.sar.

5. Execute the pack_all.bat from the command line.

   Now that the batch files are updated, you can execute the pack_all.bat file as often as you need to create a new SAR file for your store assets.

   a. Start a command prompt session by clicking **Start** -> **Programs** -> **Command Prompt**.

   b. Change directory to the \PackageSAR\<store> directory.

      For example:

      ```
      C:\ > cd \PackageSAR\itsodemo1
      ```

   c. Execute the pack_sar.bat by typing the following command:

      ```
      C:\PackageSAR\itsodemo1> pack_sar
      ```

The SAR creation process from the command line is now complete.

12.3.2 Packaging a SAR from a GUI zip utility

We experimented with several different Windows GUI-based zip utilities including WinZip, PKZip, and PowerArchiver. One of the key features to note is the ability to preserve the uppercase of the path for the \SAR-INF directory (Store Services requires this to publish the SAR).

To package a SAR file from the local file system (PackageSAR directory tree) using PKZIP 4.0 for Windows, do the following:

1. Install PKZIP 4.0 for Windows, which is available for download at:

http://www.pkware.com

2. Create the webapp.zip.
   a. Open the <install_path>\PackageSAR\ITSODEMO1\webapp directory.
   b. Select all files in the directory, and select the sub-directories **images**, **en_US**, and **es_ES**.
   c. Right-click the selected files and select **Add to Zip**.
   d. In the Add window of WinZip, click **New...** in the Add to archive section.
   e. Browse to the `PackageSAR/ITSODEMO1` directory and enter `webapp.zip` in the File name field, and then click **Open**.
   f. In the Add window of WinZip, from the Action pull-down, select **Move files**.
   g. Click the **Move** button in the top-right corner of the Add window of WinZip.

      The selected files will be transferred into a new archive, webapp.zip.
   h. Close the WinZip window.
3. Create properties.zip.
   a. Open the PackageSAR/ITSODEMO1/properties directory.
   b. Select all files in that directory.
   c. Right-click the selected files and select **Add to Zip**.
   d. In the Add window of WinZip, click **New...** in the Add to archive section.
   e. Browse to the PackageSAR/ITSODEMO1 directory and enter **properties.zip** in the File name field. Click **Open**.
   f. In the Add window of WinZip, in the Action pull-down, select **Move files**.
   g. Click the **Move** button in the top right corner of the Add window of WinZip. The selected files will be transferred into a new archive, properties.zip.
   h. Close the WinZip window.

   The contents of the directory should now resemble the directory displayed in Figure 12-1.

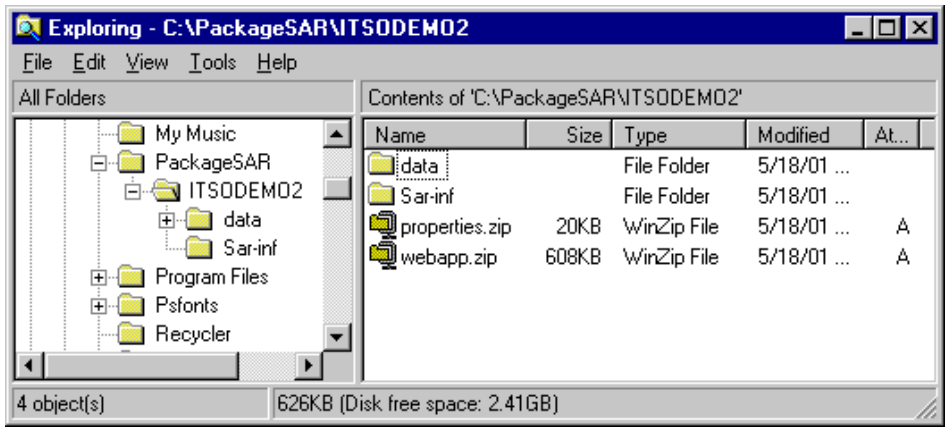


Figure 12-1 Contents of packaging a SAR for the sample ITSODEMO1

4. Create SAR containing the webapp.zip, properties.zip, data and sar-inf directory.
   a. Select all files in the PackageSAR/ITSODEMO1 directory.
   b. Right-click the selected files and select **Add to Zip**.
   c. In the Add window of WinZip, click **New...** in the Add to archive section.
   d. Browse to the PackageSAR/ITSODEMO1 directory and enter `ITSODEMO1.sar` in the File name field, and then click **Open**.
   e. In the Add window of WinZip, from the Action pull-down, select **Move files**.
   f. Click the **Move** button in the top-right corner of the Add window of WinZip.

      The selected files will be transferred into a new archive.
   g. Close the WinZip window.

Note: The SAR-INF directory within the SAR file must be uppercase for Store Services to publish the file correctly

We found PKZip for Windows and PowerArchiver very useful, in that they had the ability to support uppercase paths.

The SAR creation process from WinZip is now complete.

12.4 Deploying a SAR

After packaging a SAR, we need to verify that the procedure for packaging the store archive is working properly. To verify the SAR we need to deploy the SAR by one of the following methods:

- Deploy a SAR from Store Services
- Deploy a SAR from command line

Refer to Chapter 15, “Deployment and catalog data management” on page 425 for detailed information on store deployment.

12.4.1 Deploy a SAR from Store Services

To deploy or publish a SAR from Store Services, complete the following steps:

1. Copy the new archive ITSODEMO1.sar to the <wcs_install_path>\stores\web directory, overwriting the existing file if necessary.
2. Ensure that you have Site Administrator or Store Administrator access.
3. Open **Store Services.**

Note: If you experience problems (compilation errors) deploying your store using Store Services, we recommend that you restart Store Services.

You can also try pressing the F5 key to reload the page.

4. From the Store Archive list, select the store archive you wish to publish.

   In our example this is ITSODEMO1.sar
5. Click **Publish**.

   The Publish Store Archive window displays.
6. Select your desired publishing options. For more information on publishing options, click **Help**.

   We recommend, when publishing for the first time, that you create a fully functional store by selecting all the publishing options, including the product data option.
7. You may be prompted to confirm this step. In this case click **OK**.

Important: When we tried to publish only the updated store properties, we were prompted to enter a new path for those assets. We were forced to publish the whole store in order to maintain the existing directory structure.

8. Select **OK**.

   While the store publishes you are returned to the Store Archive list page. The publishing state is reflected in the Publish status column. Click **Refresh** to update the status.

   Tip: You may find it helpful to open the Windows NT Task Manager to monitor the progress and determine when publishing is actually finished.

9. Select the store archive from the list and click **Publish Summary** to see the results of the publish.

Attention: At this stage do not click **Launch store** yet. Close Store Services and clear cache prior to verification.

- If you change the Web application Web path or the Web application document root, you must ensure that they match the paths defined in the WebSphere Commerce Server.
- Only one store archive at a time can be published. Concurrent publishing is not supported and causes the publish of both stores to fail.

12.4.2 Deploy a SAR from command line

We used the publishstore.bat (see Example 12-8), located in your <wcs_install_path>\bin directory, to publish a SAR via the command line.

Example 12-8 publishstore.bat

```
@echo on
call setenv.bat
set
CLASSPATH=%WAS_HOME%/lib/ujc.jar;%WCS_HOME%/lib/payment/sslight-d11-rsa-rc4.zip
;%WCS_HOME%/lib/payment/eTillxml4j209.jar;%WCS_HOME%/lib/payment/etillCal.zip;%
WCS_HOME%/lib/wcssfc.jar;%WCS_HOME%/lib/wcsruntime.jar;%WCS_HOME%/lib/wcsloggin
g.jar;%WCS_HOME%/lib/wcsdevtools.jar;%WCS_HOME%/lib/wcsejbimpl.jar;%WCS_HOME%/l
ib/wcsconfig.jar;%WCS_HOME%/lib/xml4j.jar;%WCS_HOME%/properties;%CLASSPATH%

%JAVA_HOME%/bin/java com.ibm.commerce.tools.devtools.instantiation.InstUtil
-SAR %1 -svr %2 -userid %3 -pwd %4 -MODE %5 -CONFIG %6 -XML %7 -ASSET %8
```

To publish a store archive using the command line, perform the following steps:

1. Ensure that you have Site Administrator or Store Administrator access.
2. Open a command prompt and switch to the <wcs_install_path>\stores\web directory.

   Verify that the store archive file you wish to publish is located in that folder.

3. Open <wcs_install_path>\bin\publishstore.bat with a text editor.

   Note: Before you apply any changes to the file, we recommend making a backup copy.

4. Edit the following command, using the valid parameters for your store archive:

   ```
   java InstUtil -SAR sarName -SVR hostname -USERID logonId -PWD logonPwd
   -MODE {insert|update} -CONFIG configFile -XML {ALL|NOCATLG}
   -ASSET{destination1=warfile1,destination2=warfile2}
   ```

   Find parameters and their descriptions in Table 12-3.

Table 12-3 publishstore.bat parameters and description

Parameter Name	Description / Value
SAR	The name of the SAR without path specifications. In our example this is `ITSODEMO.sar`
SVR	Hostname of the WCS server, does not have to be fully qualified. In our example this is `m23bzzpg`.
USERID	Logon ID for the WebSphere Commerce Suite. In our example this is `wcsadmin`
PWD	Password for the above user ID. In our example this is `also wcsadmin`
MODE	`insert` or `update` Use `insert` when you publish your SAR for the first time or to overwrite your existing store's assets. Use `update` to update your store's assets. In our example we used `insert`.

CONFIG	The XML configuration file of the WCS instance. Provide the full path here like this:
	`<wcs>\instances\<your_instance>\xml\<your_instance.xml`
	In our example this is:
	`c:\websphere\wcs\instances\demo\xml\demo.xml`

XML	The list of XML files in the SAR to be published. To publish all, use `ALL`. To publish everything except for the catalog, use `NOCATLG`. In our example we used `ALL`.

Parameter Name	Description / Value
ASSET	The list of asset files in the SAR. For example, webapp.zip and the paths to which they will be published. For multiple asset files, list them all, separated by a *comma*. In our example we published webapp.zip and properties.zip: `c:\websphere\wcs\stores\web\=webapp.zip,c:\websphere\wcs\stores\properties=properties.zip`

Example 12-9 displays a sample edited **`InstUtil`** command for the ITSODEMO1.sar.

Example 12-9 Sample edited InstUtil command within the publishstore.bat

```
%JAVA_HOME%/bin/java com.ibm.commerce.tools.devtools.instantiation.InstUtil
-SAR ITSODEMO1.sar -svr m23bzzpg -userid wcsadmin -pwd wcsadmin -MODE insert
-CONFIG c:\websphere\wcs\instances\demo\xml\demo.xml -XML all -ASSET
c:\websphere\wcs\stores\web\=webapp.zip,c:\websphere\wcs\stores\properties=prop
erties.zip
```

5. Save and close publishstore.bat.
6. In the command prompt, type `publishstore` and then press Enter.

Attention: At this stage *do not* launch the store yet. Close the command window and clear the cache prior to verification.

Only one store archive at a time can be published. Concurrent publishing is not supported and causes the publish of both stores to *fail*.

12.5 Verify store

In this section, we describe how to clear the cache and test your store.

12.5.1 Clear cache

When verifying a store, we recommend that you clear the WCS, WAS, and Web browser cache prior to testing your store.

This section is organized as follows:

- Maintain the cache
- WebSphere Commerce Suite cache
- WebSphere Application Server cache

- Web browser cache

Maintain the cache

Maintaining the cache consists of removing files from the cache and removing records from the CACHLOG database table. How frequently you maintain the cache depends on how often updates are made to your store pages and to the database. If your store updates the catalog frequently, then files in the cache will need to be deleted more often. You may want to manually remove files from the cache, rather than allowing the cache cleanup worker to delete them, if many database records are changed at one time. Note that if you are using the **CacheDelete** command to remove files from the cache, AutoPage Invalidation must be enabled.

Under the following conditions you should manually delete items from the cache:

- JSP templates
  - Remove files from the cache using either of the following methods:
    - Use the **CacheDelete** command.
    - Manually remove the files from the cache directories.
- Registry changes

  To refresh the registry and delete the stale pages, do the following:

  a. Use the **RefreshRegistry** command, then:

  b. Remove files from the cache using either of the following methods:
     - Use the **CacheDelete** command.
     - Manually remove the files from the cache directories.

Note: If you access a store for the first time after cleaning up all caches or precompiling all JSPs you may get an error message saying the page cannot be displayed. Simply click the **Refresh** button in your browser to display the page.

WebSphere Commerce Suite cache

Before you start testing your store, make sure you clean up your WCS cache.

The WebSphere Commerce Suite cache is located in the following folder:

```
<WCS>\instances\<your_instance>\cache
```

You should delete the content of this folder including all subfolders. You may also want to turn off the WCS cache.

To turn off the WCS cache, do the following:

1. Open WCS Configuration Manager
2. Expand the tree: **Websphere Commerce Suite** -> **<your_hostname>** -> **instances** -> **<your_instance>** -> **cache**
3. Deselect Cache enabled
4. Click **Apply**
5. Restart the WebSphere Commerce Server - <wcs_instance>

WebSphere Application Server cache

Before you start testing your store, make sure you clean up your WAS cache.

The WebSphere Application Server cache relating to WebSphere Commerce Suite stores is located in the <WAS>\temp\default_host\WCS Store folder.

You should delete the content of this folder including all subfolders.

Web browser cache

Before you start testing your store, make sure you clean up your Web browser's cache. Previously cached pages or images may prevent your browser from showing the updated pages correctly.

12.5.2 Testing your store

Before you start testing your store, make sure you restarted your WCS instance in order to make all changes become effective.

To verify your store complete the following:

1. Open Store Services
2. Check the ITSODEMO1.sar
3. Click **Publish summary** in the right-hand pane.
4. Click **Launch store**
5. The ITSODEMO store should open in a separate browser window. The title of the store is still InFashion.

   It may take several minutes to compile the requested JSP. We recommend opening your Windows NT Task Manager to monitor the progress.

6. Browse the stores pages to review your changes.

Tip: To access the store's home page directly for future tests, we recommend you bookmark the store's home page.

12.6 Where to find more information

- WebSphere Commerce Suite V5.1 online help
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161
- *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*
- WebSphere Commerce Suite homepage tech library found at:

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

13

Multicultural enablement

This chapter contains information about multicultural enablement in general and in the context of the new features implemented in WCS V5.1.

The chapter is organized in the following sections:

- Overview
- Multiculture-sensitive parts of a store
- WCS V5.1 multicultural enablement
- Template management
- Multicultural programming model
- Sample store multicultural features overview
- A working example
- Tips for multicultural enablement
- Where to find more information

Note: Please keep in mind that this redbook only covers *basic* customization procedures. For *more advanced* customization refer to the *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174.

© Copyright IBM Corp. 2001

13.1 Overview

Multicultural support or enablement can be broken down into the following three subsections:

- Translation

  For example, text messages, images, menu structures, error messages, and help messages.

- Encoding

  For example, UNICODE UTF-8, ASCII, and code pages.

- Cultural specifications

  For example, currency, pricing, time, date, data-format (address).

Most of the culture-specific parts of a store are accessed and controlled in WCS by the combination of locales and language IDs.

The <language>_<locale> combination represents the display format, as it appears in the LOCALENAME column of the LANGUAGE table in the WCS database.

Multicultural enablement is embedded directly into WCS V5.1 architecture. If you want to add something new, it is not necessarily required to change the database schema. WCS V5.1 uses Unicode UTF-8 encoding.

13.2 Multiculture-sensitive parts of a store

This section describes the culture-sensitive parts of a store to provide a better understanding of the sections to follow. For example, culture-sensitive characteristics of a store can be language, data format, address details, currency, shipping, tax or taxation, page and catalog content, payment options or details and of course pricing.

13.2.1 Language

National language support is absolutely crucial when creating a store for shoppers with different languages. This has become fundamental in our global economy for companies engaged in international commerce.

WebSphere Commerce Suite offers built-in support for the following national languages:

- English for the United States

- French for France
- German for Germany
- Italian for Italy
- Spanish for Spain
- Portuguese for Brazil
- Simplified Chinese for China
- Traditional Chinese for Taiwan
- Korean for Korea
- Japanese for Japan

WCS is available in each of these languages (including software, user interfaces and a sample store.

13.2.2 Currency

National currency support is also crucial when creating a store for shoppers from different locations and currencies.

Keep in mind that there may be a need to display prices in more than one currency. Several countries within the European Union are legally obliged to display prices in their local currency as well as in euros.

Merchants can specify prices in each currency and display multiple currencies at the same time. WCS has the ability to attempt currency conversion if a price for a specific currency cannot be located.

A shopper can switch the preferred currency whenever he wishes to, even in the middle of the shopping flow. WCS will even update the price of individual items in each pending order to reflect the price in the new currency. The current preferred currency is stored in the session and will also be stored in a registered shopper´s account.

For more information on how to enable dual display for currencies, please refer to 13.7.6, “Enable counter currency/dual currency display” on page 397.

13.2.3 Data format and representation

Some example of data formatting and representation include:

- Kilograms versus ounces
- Sizes of clothing

- Centimeter vs inches
- Calendar
- Date display (for example, order of day, month and year: 20.12.2000 vs 2000/12/20 or 12/20/2000)

13.2.4 Address format

The formats of address books used in different countries may also differ. As a direct consequence, a multicultural store must be capable of displaying the proper format used for input fields (for example, during registration, as well as for output fields during checkout).

A similar issue is the telephone relationship between countries and area codes.

13.2.5 Catalog content

A merchant may be constrained by local law from selling specific products in one country, although there may be no restrictions for that product in other countries. Catalog content includes product and category descriptions, attributes and images. The language of the catalog content has to be considered in addition to the wording and spelling used for each language.

13.2.6 Page design

WCS offers merchants the ability to create separate display pages for different cultures to customize the site to meet cultural needs. A page in China may have a different “look and feel” from the one shown in Germany, although they are displaying the same item.

The WCS content includes JSPs with HTML content and images. The look and feel of the page design, the use of colors, and the page layout used must be considered. Static text on the pages and images must be translated properly into the correct language.

13.2.7 Pricing

The price of a certain product may be set differently depending on the following:

- Where the store is located (that is, where a customer makes the purchase).
- Where the customer is located (for example, where the product is shipped).
- Legal restrictions (for example, rebates and discounts applicable).

- Shopper/member group (for example, separated by demographic factors such as age, purchasing power, income, marital status, gender, affiliation to socio-economic groups, and so on).

13.2.8 Taxation

The taxation of a certain product may be set differently depending on the following:

- Where the store is located (that is, where a customer makes the purchase).
- Where the customer is located (for example, where the product is shipped).
- Legal restrictions (for example, rebates and discounts applicable).
- The product itself (for example, taxes on luxury goods may be higher than on food).

To help implement taxation, WCS includes the taxation wizards, which allow the definition of rules based on jurisdiction. Also, the data model in WCS V5.1 has been improved to allow a better integration with third-party taxation software.

13.2.9 Shipping

The shipping methods and shipping costs of a certain product may differ depending on:

- Where the store is located (that is, where a customer makes a purchase).
- Where the customer is located (for example, where the product is shipped).
- Availability of certain payment methods.
- The product itself (for example, a file download requires other “shipping” than would a nuclear power plant).

WCS includes a shipping wizard to allow the definition of rules based on jurisdiction. The concept of fulfillment centers has been added to allow for a clearer definition of ship-to and ship-from costs.

13.2.10 Payment

The payment methods may differ depending on the following:

- Where the store is located (that is, where a customer makes a purchase).
- Payment method where the customer is located (for example, where the product is shipped may use different payment methods; some European countries use a direct debit system.)

- What the merchant is willing to offer (due to bank charges, hardware cost, etc.).
- The cost of the purchase (minimum order value).

13.3 WCS V5.1 multicultural enablement

This section describes the basic features for multicultural support built in to WCS V5.1. Table 13-1 displays the corresponding locales and language IDs for the WCS supported locales.

Table 13-1 Locale and language IDs

Name	Locale	Language ID
English for the United States	en_US	-1
French for France	fr_FR	-2
German for Germany	de_DE	-3
Italian for Italy	it_IT	-4
Spanish for Spain	es_ES	-5
Portuguese for Brazil	pt_BR	-6
Simplified Chinese for China	zh_CN	-7
Traditional Chinese for Taiwan	zh_TW	-8
Korean for Korea	ko_KR	-9
Japanese for Japan	ja_JP	-10

13.3.1 Multicultural support for the Commerce Suite tools

Using Store Services, you can easily switch the language of the tools offered by WCS.

WCS requires Microsoft Internet Explorer 5.5 (we recommend SP1) to guarantee full functionality. There is no such restriction for shopper/customer Web browser accessing a WCS store.

13.3.2 Overall multicultural approach

With WebSphere Commerce Suite functionality, multicultural enablement is easier because most of the work has been automated and embedded with the logic of the product. Depending on the preferred language of the customer, static and dynamic text is retrieved in the correct language from either properties files or from the database, respectively. Data such as dates, times, measurements or currencies is formatted automatically and correctly, based on customer preferences, at retrieval time.

When a customer visits a WCS V5.1 e-commerce Web site, culturally neutral JSP templates automatically include the appropriate page components, along with retrieved text and data, to render a culturally specific view at runtime.The result is that many of the changes - multicultural look-and-feel and page content changes - necessary from one customer view to the next become transparent to the merchant. It all happens automatically with a single code base and a single set of JSP templates. Figure 13-1 depicts a customer using a multicultural-enabled WCS V5.1 e-commerce Web site.

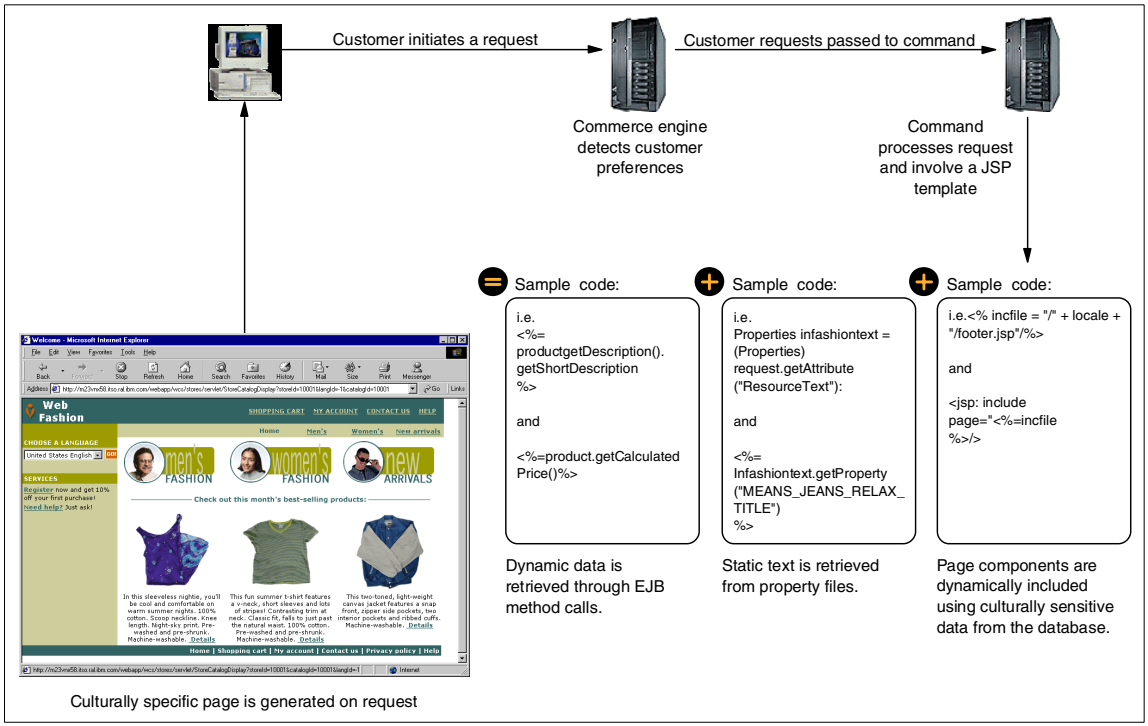


Figure 13-1 WCS V5.1 multicultural-enabled e-commerce Web site

13.3.3 JavaServer Pages (JSPs)

JSPs are at the heart of the WCS multicultural support functions. JSPs templates are used in WCS to do the following:

- Include whole page components dynamically using include statements (for example, to include a complete header).
- Retrieve static text from locale-specific text-based properties files (for example, to show labels for entry fields in different languages).
- Retrieve dynamic data (for example, the calculation of taxes).

In order to take full advantage of the benefits provided in WCS, we recommend that you keep your JSPs as language neutral as possible.

13.3.4 Properties files

Properties files form an integral part of WCS's multicultural support. They contain a locale-specific collection of Java objects in plain-text ASCII format. Potential elements of properties files are as follows:

- Text to represent the content.
- Labels (for example, used as form field labels).
- Messages (for example, errors, confirmations, and status messages).
- Alternate text for embedded objects (for example, pictures or Java applets such as IBM HotMedia applets).

Pros and cons

Table 13-2 provides a summary of the pros and cons of properties files.

Table 13-2 Pros and cons of properties files

Pros	Cons
No compilation needed	Perform ance inferior to resource bundles
Easy to maintain/plain text files/no need for recompilation	Text has to be in ASCII format, so conversion into ASCII format may be needed. This can be done by using the native2ascii program provided in IBM Developer Kit, Java Technology Edition.
To make changes effective, WAS does *not* need to be restarted	To make changes effective, clearance of WAS cache is needed.

Properties files are packaged within the properties.zip of the store archive file (SAR).

Translation of properties files

When translating properties files, consider the following issues:

- You do not need to translate any comment lines (for example, any line that begins with the pound sign #). You may translate these for your own purposes, but they will have no effect on the generated outcome.

  Note: Do not translate any content on the left side of an equal sign (=). If this value is changed from one properties file to another, the system will not recognize the two as representing the same property.

  For example, if the properties file for US English contains the line:

  ```
  lastName.Label=Last Name
  ```

  Translate it to Canadian French ("Nom de famille") as follows:

  ```
  lastName.Label=Nom de famille
  ```

  Note: Only the text to the right of the equal sign has been translated.

- Do not translate yes or no values for displayed and required attributes. In other words, you can change the values of the fields from "yes" to "no", or the other way around, but do not change it to any other value. If you write the words in any other language, the system will not recognize the values.

  For instance, if you were translating a file containing the following line:

  ```
  firstName.Displayed=yes
  ```

  Your translated file should contain either as follows (regardless of the language to which the file is being translated):

  ```
  firstName.Displayed=yes or firstName.Displayed=no
  ```

- For option attributes, translate values to the right of the semi-colon (;) only. The value to the left side must remain consistent from language to language in order for the system to recognize the options as equivalent.

  For example, if the properties file for US English contains the property:

  ```
  title.Options=MR;Mr.|MRS;Mrs.|MS;Ms.
  ```

  You could translate it to Canadian French as follows:

  ```
  title.Options=MR;M.|MRS;Mme.|MS;Mlle.
  ```

  Note: Only the text to the right of each semicolon has been translated.

  If the list included has a "yes" or "no" option, these should be translated. For example:

```
publishPhone.Options=Y;Yes|N;No
```

could be translated to Canadian French ("Oui" and "Non") as follows:

```
publishPhone.Options=Y;Oui|N;Non
```

13.3.5 Resource bundles

Resource bundles are the compiled counterpart of properties files.

Note: Only precompiled assets can become platform independent, because plain text is read from the operating system first, before being able to be accessed by Java classes. For example, a plain-text properties file in traditional Chinese will not work properly on a German operating system.

Table 13-3 provides a summary of the pros and cons of resource bundles.

Table 13-3 Pros and cons of resource bundles

Pros	Cons
Perform better than properties files	Must be compiled prior to use
Need for recompilation after changing	
To make changes effective, WAS does need to be restarted	
After compilation, resource bundles are platform independent	

13.3.6 Language ID and locale

Language IDs and locale are two separate concepts in WCS. The language ID identifies the language.

The locale describes the cultural conventions specific to a region or country. Most of the culture-specific parts of a store are accessed and controlled by the so-called locales.

The combination of language and locale represent the display format, as it appears in the LOCALENAME column of the LANGUAGE table.

Merchants can use a single WCS V5.1 store to sell goods and services internationally, and make it easier to customize dynamic data, static data, and formatting.

These three components can be associated to each LANGUAGE_ID. For each ID, the following can be specified:

- Language displayed
- Catalog and Product descriptions and images
- HTML pages and images, the so called “look and feel”
- Data formatting (that is, currency, date, measurements and address formatting)

When shopping in a WCS store, shoppers will select the preferred format (base selector). In the Web Fashion or InFashion sample store, a list box is provided in the upper-left frame to allow the shopper to switch between the languages. The Javascript behind that list box functionality changes the language ID and requests the corresponding page language ID combination. Doing this not only changes the language but also currency, product description, the currency and so on.

Shoppers have the ability to switch the format whenever they wish, even in the middle of the shopping flow.

13.3.7 WCS V5.1 uses Unicode UTF-8 encoding

The Commerce Suite database has been designed to allow you the flexibility to create and maintain internationally recognizable data by using Unicode UTF-8 encoding. UTF-8 can be converted into most international encoding formats when sent to the Web browser.

Data such as dates, currency symbols, country symbols, and language symbols conform to ISO formatting standards.

Definitions

- Character encoding form:

  Mapping from a character set definition to the actual code units used to represent the data.

- Character encoding scheme:

  A character encoding form plus byte serialization. There are seven character encoding schemes in Unicode: UTF-8, UTF-16, UTF-16BE, UTF-16LE, UTF-32, UTF-32BE and UTF-32LE

13.3.8 The sarinfo.xml in a multicultural-enabled store

Each store archive must include a sarinfo.xml file. This file, known as the *descriptor*, contains information about the store archive that is used when a store archive is published, including the names of the file asset ZIP files and the store database XML files, and the order in which they are published.

If a store archive includes files in multiple languages, the sarinfo.xml file also includes that information, and determines the order in which each language file is published.

13.4 Template management

WCS offers the following alternatives for template management to address cultural differences:

- One template for all stores
- One template per language
- One template per store

Table 13-4 provides a summary of the pros and cons of each template management solution.

Table 13-4 Pros and cons of template management solutions

Alternative	PROS	CONS
One template for all stores and all languages.	Good, when look and feel for store/language are similar.	Maintenance of properties files.
One template per language.	No properties files required. Separate directory structure.	Different templates must be created, maintained, and stored.
One template per store.	JSP templates are shared within the store.	JSP include files are required. JSP templates are exclusive for each store.

13.4.1 One template for all stores and all languages

This model is suitable where the look and feel for each store and each language are very similar. The need to maintain *only one set* of JSP templates can be seen as the main advantage, but it must manage a series of properties files. This method is the template management model and it allows for site-wide page-design changes since only one template needs to be changed.

For example, if you have two store locations, each displaying US English and Canadian French, you might organize your JSP templates as follows:

```
<wcs_install_path>/stores/web/<your_store>/your_sample.jsp
```

The path for the properties files for this JSP template would be stored as follows:

```
<wcs_install_path>/stores/web/<your_store>/properties/en_US/your_sample.pro
perties
<wcs_install_path>/stores/web/<your_store>/properties/fr_CA/your_sample.pro
perties
```

In this case, when registering the JSP files, only the file type needs to be included in the file registry.

Using this method, only one set of JSP files needs to be registered for all stores and all locales. Here, the properties files must be stored separately because they contain culturally sensitive information, whereas the template itself, which is completely neutral, is stored in a common directory.

13.4.2 One template per language

To use this model, a separate directory must be created within each store's template directory for each language supported by that store. A different template must be stored within each of these language-dependent directories. No properties files are required in this model, since each store and locale combination has its own JavaServer Page template.

For example, if you have two store locations, each displaying United States English and Canadian French, you might organize your JSP templates as follows:

```
<wcs_install_path>/stores/<your_1st_store>/web/template/en_US/your_sample.jsp
<wcs_install_path>/stores/<your_1st_store>/web/template/fr_CA/your_sample.jsp
<wcs_install_path>/stores/<your_2nd_store>/web/template/en_US/your_sample.jsp
<wcs_install_path>/stores/<your_2nd_store>/web/template/fr_CA/your_sample.jsp
```

In this case, when registering the JSP template, the locale and file type will have to be included in the file registry. Each store and each locale must have a complete set of registered templates.

13.4.3 One template per store

JSP templates for this model are shared within a store, but are exclusive to a single store. JSP include files are required in this model to allow for template sharing between each language format.

For example, if you have two store locations, each displaying United States English and Canadian French, you might organize your JSP templates as follows:

```
/webapp/StoreA/web/template/abc.jsp
/webapp/StoreB/web/template/abc.jsp
```

The path for properties files within this template management model would resemble the example below:

```
/webapp/StoreA/web/properties/en_US/abc.properties
/webapp/StoreA/web/properties/fr_CA/abc.properties
/webapp/StoreB/web/properties/en_US/abc.properties
/webapp/StoreB/web/properties/fr_CA/abc.properties
```

When registering the JSP templates for this model, only the file type needs to be included in the file registry. Each store will have to register its own complete list of the JSP files.

13.5 Multicultural programming model

WCS V5.1 is designed to enable a single store to work with any language desired. To benefit from all implemented features, developers need to follow a certain programming model.

Figure 13-2 shows the multicultural programming model for the “’one template for all stores” model used by the sample store:

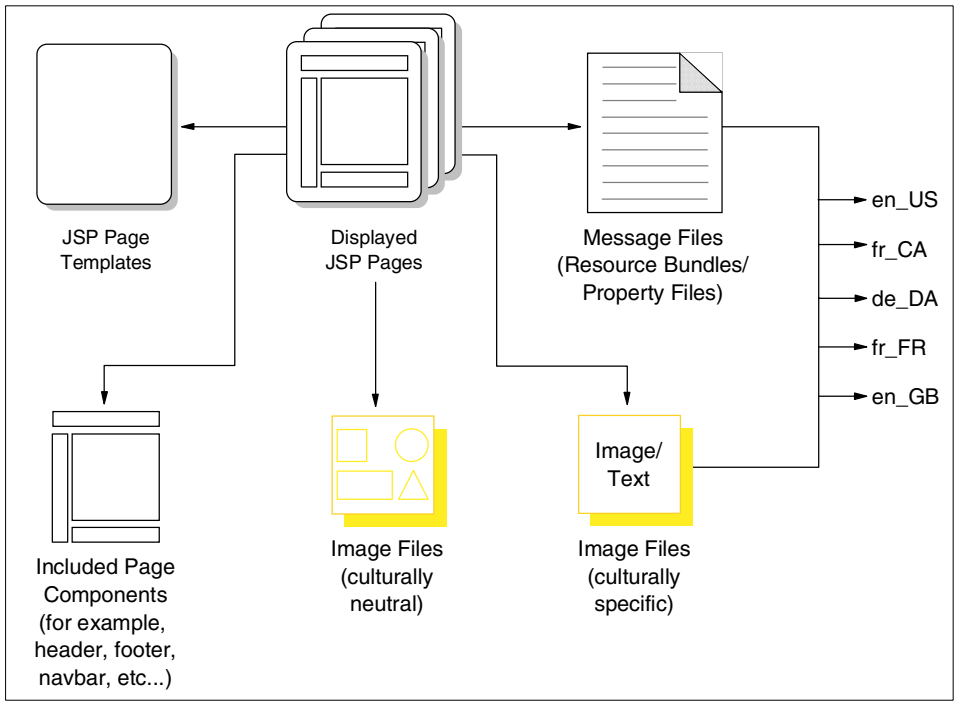


Figure 13-2 Multicultural programming model

13.5.1 Multicultural page framework

This section describes the multicultural page framework.

Retrieving static HTML pages and images

Static HTML text will be stored in directories that will be qualified by a locale. At runtime, the page will be retrieved depending on which language the shopper is using.

```
<Store Document Root>/en_US/page.html
<Store Document Root>/fr_FR/page.html etc...
```

Multilingual static text

All static text on a JSPs will be retrieved from the properties files. For example:

```
ResourceBundle myResource = ResourceBundle.getBundle('messages', 'locale' );
```

Where the properties files would be:

- messages_en_US.properties
- messages_fr_FR.properties

Dynamic data

Dynamic data is retrieved using various server bean methods. Retrieval of data is automatically performed in the context of the current language.

For example:

```
<%String languageId = request.getParameter("lang");ProductDataBean product =
new ProductDataBean();DataBeanManager.activate(product, request);%>

<p>Product description here: <%= product.getDescription().getShortDescription()
%></p>
```

13.5.2 Data storage

To support multiple languages in a product, it is necessary to be able to store data from each language in a single database. Data is stored in the database using UTF-8 encoding. WCS configures the database at creation time to store data in UTF-8. A UTF-8 database has the correct character mappings to handle all single-byte, double-byte, and bidirectional characters (such as Hebrew).

JDBC drivers load data from the database and store textual data in java.lang.String. To do this, the drivers convert text data from UTF-8 to Java's native 16-bit UNICODE encoding.

13.5.3 Data input

Multilingual data can be entered through the browser using an IME (Input Method Editor). This is available for most browsers and/or operating systems.

Each LANGUAGE_ID in the language table maps to an encoding value. This value is used to interpret the data going out to the browser as seen in Table 13-5.

LANGUAGE_ID	LOCALENAME	LANGUAGE	COUNTRY	VARIANT	ENCODING
-1	en_US	en	US	-	ISO8859-1
-3	de_DE	de	DE	-	ISO8859-3

Table 13-5 Data input overview

Based on the current LANGUAGE_ID, WCS will properly convert the data coming in from the browser into 16-bit UNICODE using the setCharacterEncoding() method.

From there, the data is sent to the database, and then the data transformation from 16-bit UNICODE to UTF-8 (8-bit UNICODE) and is stored in the database.

13.5.4 Data output

Strings are handled inside Java code as UNICODE 16-bit encoding. JSPs output strings using UNICODE 16-bit encoding (because JSPs are 100% Java code). JSP authors can specify the encoding to be used to send the HTTP reply back to the browser. For example:

```
<%@ page contentType="text/html; charset=Shift-JIS"%>
```

WAS will convert the JSP output text from 16-bit UNICODE to the target encoding as specified in the JSP page and send the converted data back to the browser. For example:

```
Content-type: text/html; charset=Shift-JIS
```

The browser will interpret the HTTP reply based on the encoding specified in the header and automatically switch to the encoding setting specified to display the data.

13.6 Sample store multicultural features overview

The sample store that comes with WCS gives you a foundation on which to create your store and demonstrates how you can create and maintain a site that is suitable for a multicultural customer base.

The sample allows customers to select the language in which they would like to view the site by selecting the desired display format from a list (and thereby switching the language ID). The list is displayed in a drop-down list in the left-hand frame throughout the site. Customers can navigate through the site, viewing it in the language of their choice.

13.6.1 Programming model

The sample has been developed using a multicultural programming model that uses a single template for all languages. Each supported language is represented in its own properties file, called infashiontext.properties. It contains the translated text and messages for that language. The name of the file is the same for all languages.

For example, the file InFashion/en_US/infashiontext.properties contains information pertaining to the display of the text in US English, and the InFashion/jp_JA/infashiontext.properties file contains text in Japanese. For an alternate method of organizing properties files, see the registration sample.

Within the properties file, a number of elements of the page are translated:

- Text (textual page content)
- Labels (form field labels)
- Messages (error, status, and confirmation messages)
- Alternate text (for images, Java applets, and other embedded objects)

The format of the text is also indicated in the properties file under the ENCODESTATEMENT properties.

For example, the infashiontext_en_US.properties file contains the statement:

```
ENCODESTATEMENT = text/html; charset=ISO_8859-1
```

This indicates the character set in which the text is displayed in a browser. Because it is specified within the properties file instead of in the JSP template itself, the character set can be different for each language. The value is retrieved from the properties file, and the character-encoding of the generated JSP page is set using the following statement in JSP template:

```
<%response.setContentType(infashiontext.getString("ENCODESTATEMENT")); %>
```

13.6.2 JSP include of getResource.jsp

At runtime, each requested JSP includes the file getResource.jsp. Within this file, the command context is retrieved, and from that, the locale is used to retrieve the infashiontext properties Java object, which obtains its values from the infashiontext.properties file in the appropriate locale-specific directory. The template then has access to each of the properties as needed, through the use of the getProperty() method of the properties object.

The template uses a similar method to display translated image files, such as buttons or banners. The locale and language are retrieved at runtime to determine the correct folder in which to look for the image file. In the example above, the template might look for the file Infashion/<language_Locale>/images/go_button.gif, where<language_Locale> is replaced by the display format from the command context.

For example, the resulting page will display the image:

```
Infashion/en_US/images/go_button.gif or Infashion/jp_JA/images/go_button.gif.
```

13.6.3 Catalog content

The same method used with the getResource.jsp does not apply to the display of online catalog content. Instead, the catalog contains multiple translations, one for each locale supported. At runtime, the command context is sent through a data bean to determine which translation will be retrieved from the database and displayed on the page.

13.7 A working example

The sample store contains language-neutral JSPs that can easily handle multiple languages. The InFashion and Web Fashion sample stores are already multicultural enabled. In this example, we will demonstrate how to implement a new language.

We will implement German, although you may already have a version of the sample store supporting German. The refresh from WCS V5.1 (WCS V5.1.0.1) includes multicultural SARs for all of the 10 automatically supported languages by WCS. We want to show you the overall procedures, make you familiar with the tools to use, and thereby enable you to add new languages, special dialects or sublanguages that don't come with WCS automatically.

Note: Creating a complete multicultural store from scratch is out of the scope of this redbook.

We used the InFashion and Web Fashion sample stores to demonstrate some of the new features in WCS V5.1 for multicultural enablement.

13.7.1 Creating a multicultural-enabled store

In this section we highlight how to implement an additional language (in this case German) and multicultural assets in the Web Fashion sample store.

To add a new language to a store, do the following:

1. Ensure that the language is available to your site. If it is *not* one of the 10 national languages officially supported by WebSphere Commerce Suite, you will have to create a new display format for the language. Refer to WCS online help on how to create a new display format.
2. Use the Store Profile Editor (via Store Services) to add the language to the list of those supported by the store.
3. Translate the catalog content and load the data to the Commerce Suite database using the Commerce Suite Loader Package.

4. Translate Web assets, such as banners and buttons, following the multicultural programming model described above.
5. Translate any properties files or resource bundles used by JSP templates, following the multicultural programming model.

Table 13-6 provides a summary of the major steps to take to add a language.

Table 13-6 High level steps to add a language

Step/Action	Description/Comments	Mandatory (yes/no)
Choose template and programming model	Important for the overall store creation and development approach	yes
Add or change language support on instance level	Mandatory in order to create a fully functional instance	yes
Add additional language to the store	Using Store Services	yes
Create multicultural assets (for example, different catalog or images for new language)	We will use the same products, just translate the descriptions	yes
Check the language table		no
Update/check the language description (LANGUAGEDS) table		yes
Clean up cache		yes
Restart WCS instance		yes
Verify store		no

13.7.2 Add/change language support on instance level

There are several manual steps to take prior to creating a multicultural-enabled instance.

Populating the database correctly

Before you create a WCS instance, you have to update the populatedb.db2.bat file in order to add support for the desired language(s), as follows:

1. Open the populatedb.db2.bat file with a text editor of your choice.

   The file resides in the <wcs_install_path>\bin folder.

2. Add a row for each language to be supported.

   The following example shows two lines, the first one for Spanish, which should already be there, and the second for German:

```
%JAVA_HOME%\bin\java -classpath %CP% %massLoader% -infile
wcs.bootstrap_multi_es_ES.xml %dbparms% -method sqlimport >> %log%
anish.

%JAVA_HOME%\bin\java -classpath %CP% %massLoader% -infile
wcs.bootstrap_multi_de_DE.xml %dbparms% -method sqlimport >> %log%
```

Note: The mentioned bootstrap files are usually located in your <wcs>\schema folder.

3. Save and close the file.

Adding/change the instance language support

The next step is to add aditional language support by completing the following:

1. Open the <your_instance>.xml file with an XML editor of your choice.

   The file resides in the <wcs>\instances\<your_instance>\xml folder.

2. Search for the SupportedLanguages entry and add the desired language ID(s).
3. Save and close the file.
4. Restart WebSphere Commerce Server - <wcs_instance>.

If an instance has already been created not using the updated populatedb.db2.bat file, you will be required to create another instance.

We will give you a short description of how to add another instance on the same machine using the Microsoft Loopback Adapter by highlighting the top-level-steps to take:

1. Open the hosts file in a text editor of your choice.

   The file is located in the <winnt>\system32\drivers\etc folder.

2. Add the hostnames and IP addresses you wish to use.

Note: The IP addresses do not have to be real. This is extremely useful for multiple instances on one machine that has only one real IP address.

Still make sure there is no IP overlapping in your network, which could cause IP address conflicts.

3. Save and close the file.
4. Now go to Network setting via Control Panel and add another adapter.
5. Select **MS Loopback** and click **OK**.
6. Follow the instructions in the window and add the second IP address used in step 2 and click **OK**.
7. Reboot your machine (mandatory).
8. For information on how to actually add the second instance, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*.

Note: Keep in mind that at least 512 MB of RAM are required. Additional instances require approximately 150 MB.

13.7.3 Translation of assets

The following files/assets can/should be translated:

- Catalog images (product description)
- HTML assets such as images (buttons), window titles, alternate text, etc.
- All kinds of text used (this is done via changing the appropriate properties files)
- Help files
- Menus
- Error messages

13.7.4 Translation of properties files

In order to introduce German as an additional language to the sample store, the appropriate properties files must be translated. Refer to “Translation of properties files” on page 383 for details.

13.7.5 Update the language description table

The language description table needs to be updated manually, as follows:

- Open DB2 command window
- Insert values into the table via a SQL statement (see Figure 13-3).

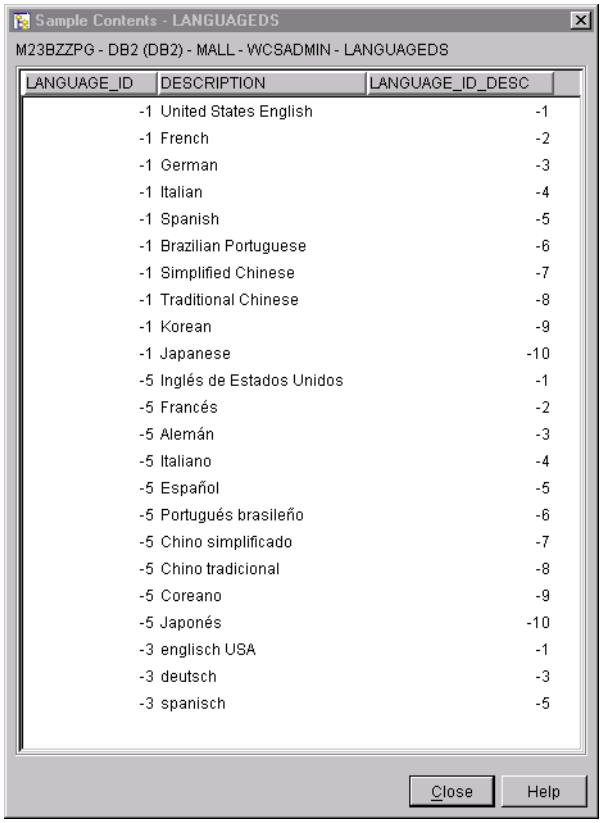


Figure 13-3 Sample content - LANGUAGEDS table

13.7.6 Enable counter currency/dual currency display

In this section we describe how to enable counter currency and dual currency display by using the Web Fashion sample store.

Thanks to WCS V5.1 already built-in features, you can easily display counter currencies and so make a store euro-ready. This ability is important for stores operating in those countries that are required by local law to show prices in their local currency as well as in euro until the euro becomes the single currency in the European Union by 1 January 2002.

Enable dual display of supported currencies

To enable dual display of supported currencies, do the following:

1. Open a database command window.

2. For every combination of supported currency and counter value currency, enter the following on one line:

   ```
   INSERT INTO CURCVLIST (STOREENT_ID,CURRSTR, COUNTERVALUECURR,
   DISPLAYSEQ)VALUES (<store_entity_id>, '<currency>',
   '<counter_value_currency>', <display_sequence>)
   ```

   Where:

   <store_entity_id> is the store or store group ID.

   <currency> is the 3-character ISO 4217 currency code representing the currency. This code must appear in the SETCCURR column of the SETCURR table.

   <counter_value_currency> is the 3-character ISO 4217 currency code representing the counter value currency. This code must appear in the SETCCURR column of the SETCURR table.

   <display_sequence> is the number that indicates the presentation order of the counter value currency. Counter value currencies are displayed in ascending order based on the counter value display sequence specified in the DISPLAYSEQ column in the CURCVLIST table.

3. Restart the WebSphere Commerce Server - <wcs_instance> or use the RefreshRegistry command.
4. Manually remove pages showing the currency from the cache.

   In our example the statement should look like this:

   ```
   db2 => insert into curcvlist (storeent_id, currstr, countervaluecurr,
   displayseq) values (10001, 'USD', 'EUR', 1)
   ```

   Make sure this confirmation message appears afterwards:

   ```
   DB20000I  The SQL command completed successfully.
   ```

13.8 Tips for multicultural enablement

This section includes tips for multicultural enablement.

Populating the database correctly

Before you create an instance, you have to update the populatedb.db2.bat file in order to add support for the desired language(s). Do the following:

1. Open the populatedb.db2.bat file with a text editor of your choice.

   The file resides in the <wcs_install_path>\bin folder.

2. Now you have to add a row for each language to be supported.

The following example shows two lines, the first one for Spanish, which should already be there, and the second for German:

```
%JAVA_HOME%\bin\java -classpath %CP% %massLoader% -infile
wcs.bootstrap_multi_es_ES.xml %dbparms% -method sqlimport >> %log%
anish.

%JAVA_HOME%\bin\java -classpath %CP% %massLoader% -infile
wcs.bootstrap_multi_de_DE.xml %dbparms% -method sqlimport >> %log%
```

Note: The mentioned bootstrap files are usually located in your <wcs_install_path>\schema folder.

3. Save and close the file.

Adding/changing the instance language support

The next step is to add additional language support by completing the following:

1. Open the <your_instance>.xml file with a xml editor of your choice.

   The file resides in the <wcs_install_path>\instances\<your_instance>\xml folder.

2. Search for the SupportedLanguages entry and add the desired language ID(s).
3. Save and close the file.
4. Make sure to restart your Commerce Server.

13.9 Where to find more information

- WebSphere Commerce Suite V5.1 online help
- *Multicultural enablement with WebSphere Commerce Suite*, white paper found at:

  http://www.ibm.com/software/webservers/commerce/wcs_pro/lit.html
- Information about Unicode UTF-8 encoding can be found at:

  http://www.unicode.org/
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174

14

Implementing auctions

This chapter describes the elements of an auction and provides an implementation example highlighting the features of WebSphere Commerce Suite V5.1 for auctions.

This chapter is organized into the following sections:

- Auctions overview
- Create a sample auction store
- Create an auction
- Auction database tables

© Copyright IBM Corp. 2001

14.1 Auctions overview

The auctioning component provided with WebSphere Commerce Suite V5.1 allows you to sell products to the highest bidder. Auctions can be implemented as part of an existing store, such as a B2C or B2B store. Alternatively, you can implement an auction-only store. With Commerce Suite, you may conduct several auctions simultaneously.

14.1.1 When to use auctions

The auction component provides an ideal environment for implementing small to moderate-scale auctioning as part of your e-commerce solution. We have listed some situations that auctions are well suited for:

- When you are uncertain about the size of the market and the willingness of buyers to purchase a product.
- When a product's price has been set too high initially, and you want to determine a price controlled directly by market demand.
- When you want to promote new product lines or liquidate inventory.

14.1.2 Types of auctions

WCS Commerce Suite Accelerator supports three types of auctions: open cry, sealed bid, and Dutch auctions.

Open cry auction

All bids are available for participants to see.

Sealed bid auction

The bidder can only view those bids that they have submitted. They cannot see bids submitted by other users. The auction administrator can see the details of all bids for each auction. Bidding takes place over a specified time period; at the end of that period, the highest bid wins.

Dutch auction

Lets the auction owner set a price that may be accepted by the bidder, if not the price may be lowered or raised.

This is not a "true" Dutch auction, as implemented in the Dutch Flower Market. A true Dutch auction uses a continuously decreasing price, similar to a countdown timer. A person clicks a button when they are willing to pay the current displayed price. This person then enters the requested quantity, and the "clock" is reset to some higher initial value to start the countdown once again.

What is offered in WCS better fits the "Bargain Basement" model. For example, prices are reduced once per week until inventory is gone. You can take it now at a higher price, or you can wait and take a chance that it will still be available next week, at a lower price.

14.1.3 Auction rules

Every auction is governed by a set of rules that the bidder must read before participating. Auction rules are established when creating an auction and include items such as these:

- The auction type
- The product name
- The quantity available
- Whether a reserve price exists for the auction
- The deposit amount to be forfeited if the winner refuses to accept the auctioned items
- The auction start date and time
- The conditions under which the auction will end, such as a scheduled end date and time
- Bid control rules include the following:
  - Minimum bid price
  - Quantity
  - Minimum quantity

    This allows the auction owner to specify a minimum quantity for the sale. For example, if you want to buy this screw, you need to buy at least 100 of them.
  - Bid increment within a price range
- Pricing mechanisms for the auction

If auction rules change during an auction, bidders must reread the rules before submitting or updating bids. Bids submitted prior to a rule change are not affected and may still win the bidding.

14.1.4 Auction style

The auction wizard helps you create an auction style, which is basically a custom template that would save time by supplying values for specified fields when you create an auction. Auction styles will be based on the above-mentioned three types of auctions. Auction styles will be most useful when you have standard and defined rules for auctions.

14.1.5 AutoBids

Participants may have bids submitted in open cry auctions automatically by setting up autobids that specify the maximum bid value and other information. This feature is also known as *proxy bidding* in other significant auction sites in the market.

The **`ProcessAutoBids`** command (based on maximum bid price, bid quantity and bid creation time) determines the current leaders and automatically submits bids for those who have autobids in order to keep them among the leaders. As a comparison, in a footrace, there is no winner until after the finish line has been crossed.

Note: Since there can be more than one item on auction, there can be more than one currently winning bid.

Autobids may also be known as order bids, because the user places an order with the system to bid on their behalf.

14.1.6 Scheduled jobs

Enabling the scheduler to run auction commands is an important first step towards creating an auction store. The following auction commands are executed periodically for the auction status to be updated:

- MonitorAuctions
- ProcessOpenCryBids
- ProcessDutchBids
- ProcessAutoBids
- DoAuctionNotify
- CompleteOrder
- FinalizeAuction

These commands start and stop auctions, determine the high bid and lowest winning bid, update the available quantity for Dutch auctions, submit bids from autobids, send automated messages to customers via e-mail, create an auction from winning bids, and determine winners for open cry and sealed-bid auctions.

Enable auction scheduler jobs

Auctions use the Job Scheduler, a component of a Commerce Server that schedules and launches jobs according to a timing framework. A job is a Commerce Suite command scheduled to run at a specified time or interval. For example, the **MonitorAuction** command is a scheduler job.

Before working with auctions for the first time, you must activate auction scheduler jobs by updating the scheduler configuration table. To activate these jobs, first modify the hostname for the following file on your platform:

Windows	\schema\wcs.auction.sql
AIX	/usr/lpp/CommerceSuite/schema/wcs.auction.sql
Solaris	/opt/WebSphere/CommerceSuite/schema/wcs.auction.sql

For DB2, type the following SQL statements in a DB2 command window, replacing <wcs_database> with your WCS database name:

```
$ db2 connect to <wcs_database>
$ db2 -tvf wcs.auction.sql
```

For Oracle, type the following from the command prompt, replacing userid and password with the user ID and password for the Oracle database:

```
> sqlplus userid/password @wcs.auction.sql
```

To enable the scheduler to run the auctions commands, do the following:

1. Update/insert into the schconfig table the entries for each of the auction commands and set their status to Active (sccactive =‘A’). For example:

   ```
   update schconfig set sccactive='A' where sccpathinfo in ('MonitorAuctions',
   'ProcessOpenCryBids', 'ProcessDutchBids', 'ProcessAutoBids',
   'DoAuctionNotify', 'CompleteOrder', 'FinalizeAuction');
   ```

2. Insert into the schstatus table entries for the auction commands. For example:

   ```
   insert into schstatus values
   (2353,2052,'I','2000-10-30-10.50.21.000000',null,null,null,null,0,0,null);
   ```

   The value 2052 in the above insert corresponds to the sccjobrefnum value in the schconfig table for the **MonitorAuctions** command.

14.2 Create a sample auction store

This section describes how to create a sample auction store.

14.2.1 Enabling the sample auction store

WCS V5.1 comes with sample auction files that can be used to quickly get a simple auction store up and running. Refer to WCS V5.1 online help and search for “Configure the sample auction” to find detailed steps to configure the sample auction store.

Once you have created and published the sample Web Fashion store, use the following instructions to open the newly created sample store archive in WebSphere Commerce Studio. For this example we call our new sample store “eAuction”. The instructions below show how to add the sample auction functionality to your eAuction sample store.

1. Start IBM WebSphere Commerce Studio V3.5 from your program menu and select **Create a New project.**
2. Choose the **Create using Store Archive** option. Enter the user ID and password to log on to your Commerce Suite Instance.
3. A window with the store archive files list appears. Select the **eAuction Store Archive** file from the Archive file list and import it to WCS Studio.
4. Right-click **eAuction** under the stores folder, and select **Insert** -> **File**.

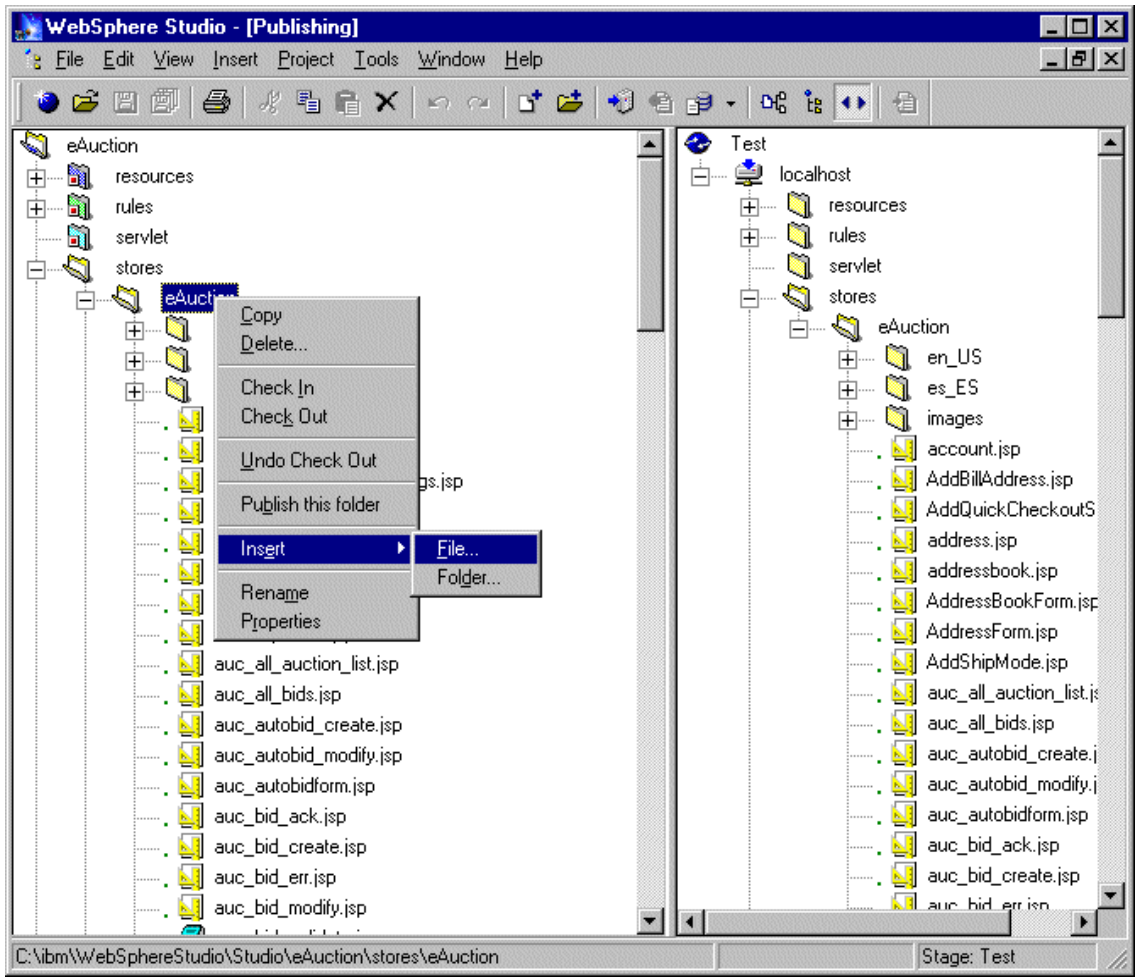


Figure 14-1 WebSphere Commerce Studio - insert file

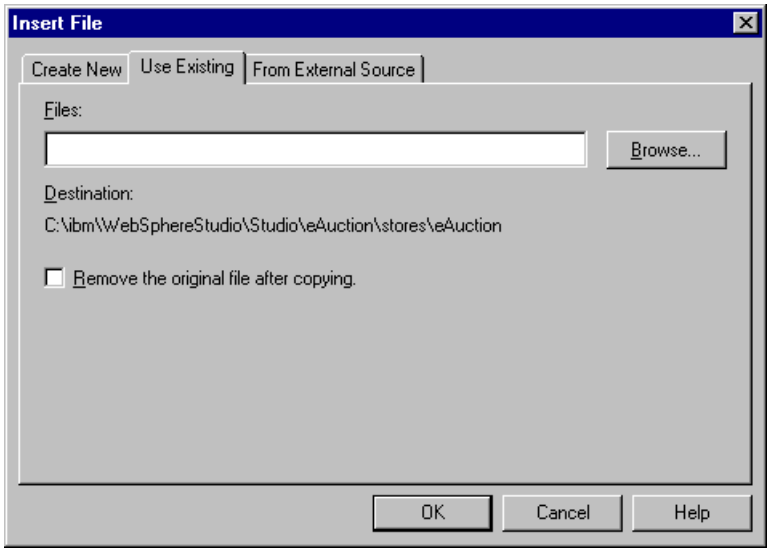


Figure 14-2 Insert file window

5. Click **Browse** in the Insert File window. On the Use Existing tab, navigate to the <wcs_install_path>\samples\web\Auction folder, select all the JSP files, and add them to your project.
6. Right-click the **Stores** folder and select **Publish this Folder**. Accept the default options, ignore the broken child links options, and click **OK**. In Commerce Suite, Java commands take the form of URLs and if WCS Studio cannot find associated files for these, it reports broken links.
7. If Publishing is successful, you can access the sample auction pages from a browser through the URL:

   `http://`*`hostname`*`/webapp/wcs/stores/servlet/AuctionSample?storeId=`*`nn`*

   Where `nn` is your store ID.

Note: An alternative way of adding the sample auction store to your eAuction store would be to manually copy the files in the following directories into the eAction.sar file by opening it using a zip utility. This is not the recommended way of modifying Web assets within a store archive.

Assuming your WCS and IBM HTTP Server install paths are c:\ibm\wcs and c:\ibm\http respectively, the assets for the auction sample will reside in the following directories:

1. Contents of c:\ibm\wcs\samples\web\Auction*.* should be copied into webapp.zip within the Store Archive
2. c:\ibm\wcs\samples\properties*.* should be copied into properties.zip
3. c:\ibm\wcs\samples\web\Auction\en_US*.gif should be copied into webapp.zip

Completion of the above steps will enable you to view the auction pages but with no data. In order to view products on auction, you need to do two things.

- Using the Commerce Suite Accelerator or Manual methods, as explained in 14.3, “Create an auction” on page 415, create auctions
- Enable the scheduler to run auction commands by following the steps explained in 14.1.6, “Scheduled jobs” on page 404

14.2.2 Add an auction to an existing store

In this section we discuss and provide steps to integrate the Web Fashion sample store and the sample eAuction store. As provided in WCS, the look and feel of these two stores are quite different. For example, the auction store uses HTML frames across all its display pages (JSPs) but the Web Fashion store does not.

In order to integrate the two sample stores do the following:

- Use the eAuction sample store that was created as detailed in 14.2.1, “Enabling the sample auction store” on page 406.
- Provide access to the eAuction store from the Web Fashion store and vice versa. The My Account Page in the Web Fashion store is an ideal place to provide this link.
- Change the eAuction store display pages to have the same look and feel as the Web Fashion store. This would involve removing the frames within the eAuction store JSPs and including header and footer to the auction pages similar to those in the NeFashion store pages
- Package the eAuction store into a SAR file.

The integrated store will be named and referred to as the eAuction store. The eAuction store will provide the same functionality as provided in the Web Fashion and eAuction sample stores. But the GUI of the eAuction store will be changed to look similar to the Web Fashion store.

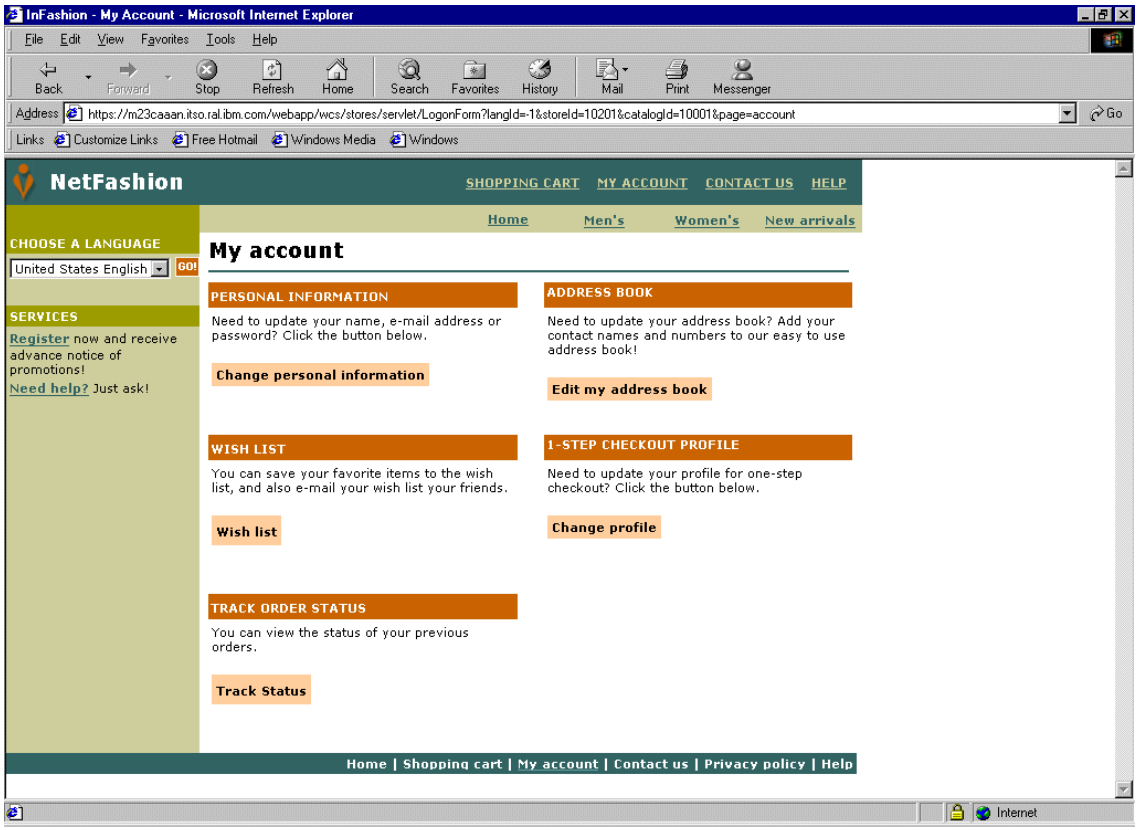


Figure 14-3 My account - before changes to Web Fashion

We will make changes to account.jsp, auc_common.jsp and infashiontext_en_US.properties files as illustrated below.

account.jsp: Add the following piece of code after the completion of the table data tag to Track Order Status.

Example 14-1 account.jsp - sample to track order status

```
</tr>
</table>
</td>
<td align="left" valign="top" width="280" class="topspace">
<table cellpadding="0" cellspacing="0" border="0">
<tr>
<td align="left" valign="top">
<font
class="text"><%=infashiontext.getString("AUCTION_HOME_MESSAGE")%></font><br><br
></td>
```

```
</tr>
<tr>
<td align="left">
<table cellpadding="4" cellspacing="0" border="0">
<tr>
<td align="left" valign="middle" bgcolor="#FFCC99">
<A
href="AuctionHomeView?storeId=<%=storeId%>&langId=<%=languageId%>&catalogId=<%=
catalogId%>"><font
class="strongtext"><%=infashiontext.getString("AUCTION_HOME2")%></font></a></td
>
</tr>
</table>
</td>
```

auc_common.jsp:This file is used as a JSP page include in all the auction JSPs. The code changes shown in Example 14-2 are intended to provide the storename and the infashion properties file to all the auction-related JSPs.

Example 14-2 auc_common.jsp -

```
String storeName = null;
......
storeName =
sdb.getDescription(aCommandContext.getLanguageId()).getDisplayName();
request.setAttribute("storeName", storeName);
.....
if (infashiontext == null)
{
infashiontext = ResourceBundle.getBundle(storeDir + "/infashiontext",
locale );
request.setAttribute("ResourceText", infashiontext);
}
```

infashiontext_en_US.properties: The changes made to the account.jsp file refer to the infashiontext_en_US.properties file for text to be displayed on the My account page. Having the textual content in a properties file is good coding practice, because it makes it easy to enable the store for multicultural use. Add the lines in the following code, with references to Auction and to the properties file.

Example 14-3

```
#account.JSP phase II
ACCOUNT_WISHLIST = WISH LIST
ONESTEP_CHECKOUT = 1-STEP CHECKOUT PROFILE
ACCOUNT_WISHLIST_MESSAGE = You can save your favorite items to the wish list,
and also e-mail your wish list your friends.
ACCOUNT_WISHLIST2 = Wish list
```

```
ONESTEP_CHECKOUT_MESSAGE = Need to update your profile for one-step checkout?
Click the button below.
ACCOUNT_CHANGEPROFILE = Change profile
TRACK_ORDER_STATUS = TRACK ORDER STATUS
TRACKORDER_STATUS_MESSAGE = You can view the status of your previous orders.
TRACK_STATUS = Track Status
AUCTION_HOME = AUCTION HOME
AUCTION_HOME_MESSAGE = You can view current and future auction as well your
bids.
AUCTION_HOME2 = AUCTION HOME
```

After the above changes have been made and the Store Archive file has been published, the My Account page will look like Figure 14-4.

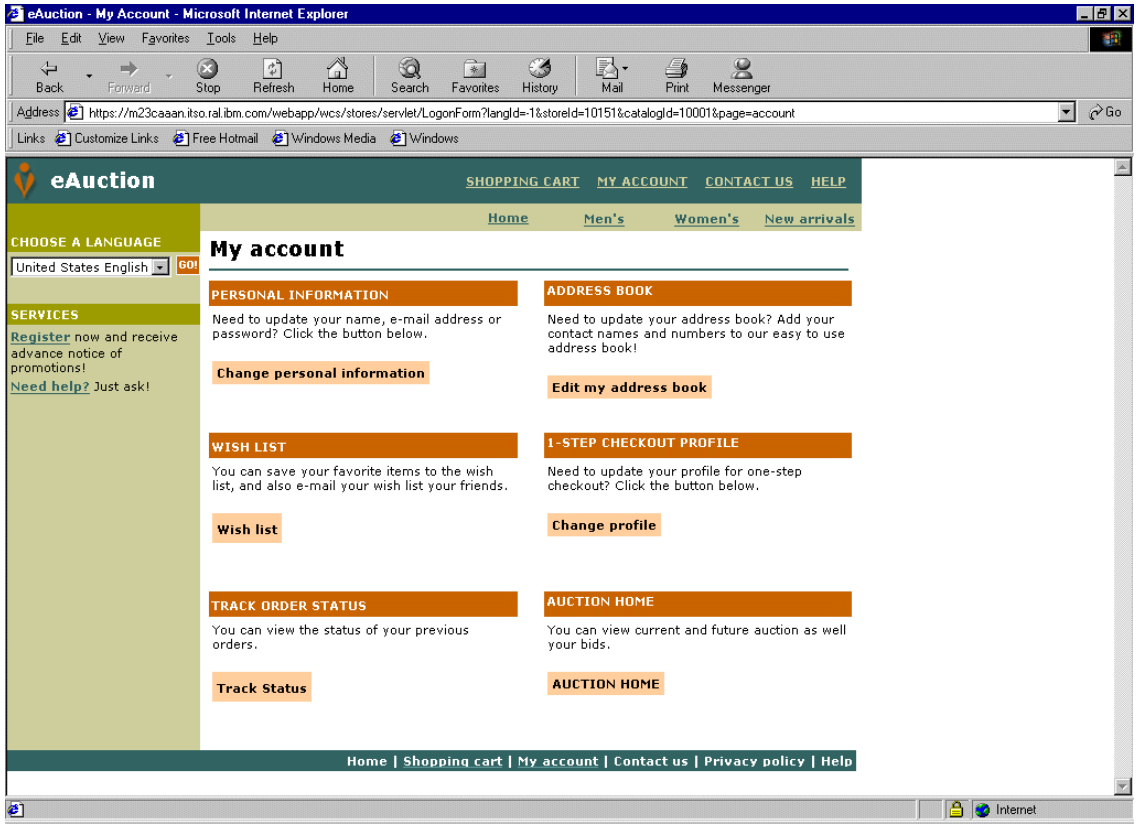


Figure 14-4 My account after changes - eAuction

Clicking the **Auction Home** link provided on the My account page will take you to the eAuction home page. Below is an image of the eAuction sample store's home page before changes.

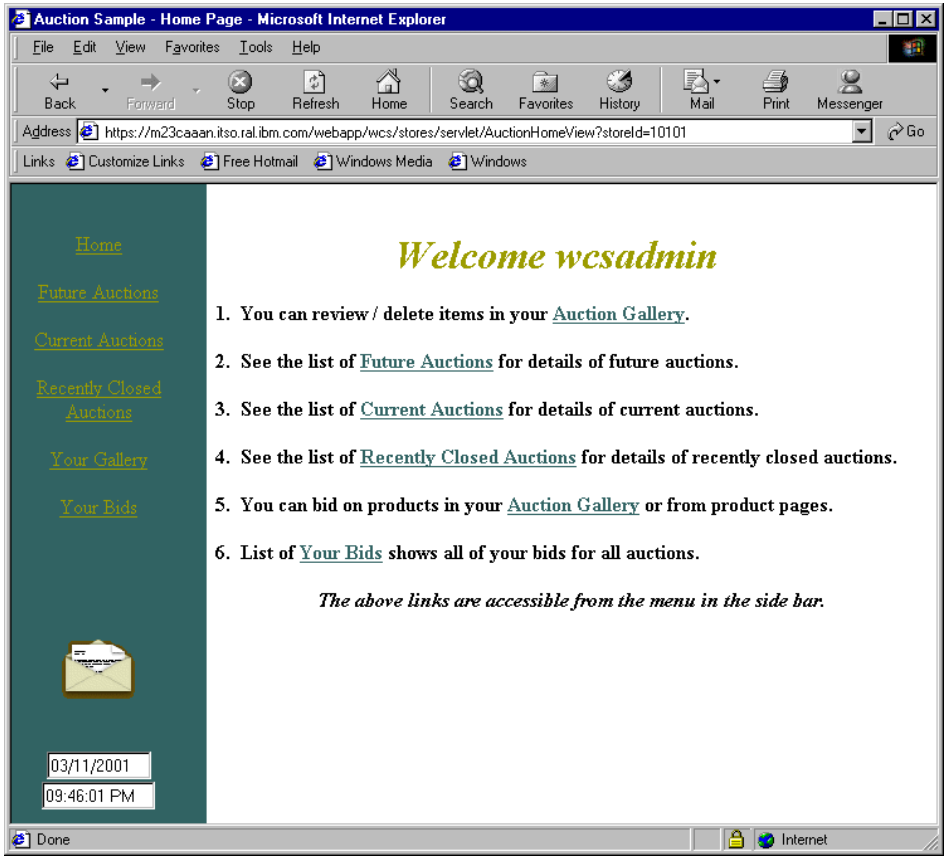


Figure 14-5 eAuction sample store home page - before changes (w/ frames)

In Figure 14-5 is an image of the eAuction store's home page. The header and footer being used in the Web Fashion store have been added to the auction pages.

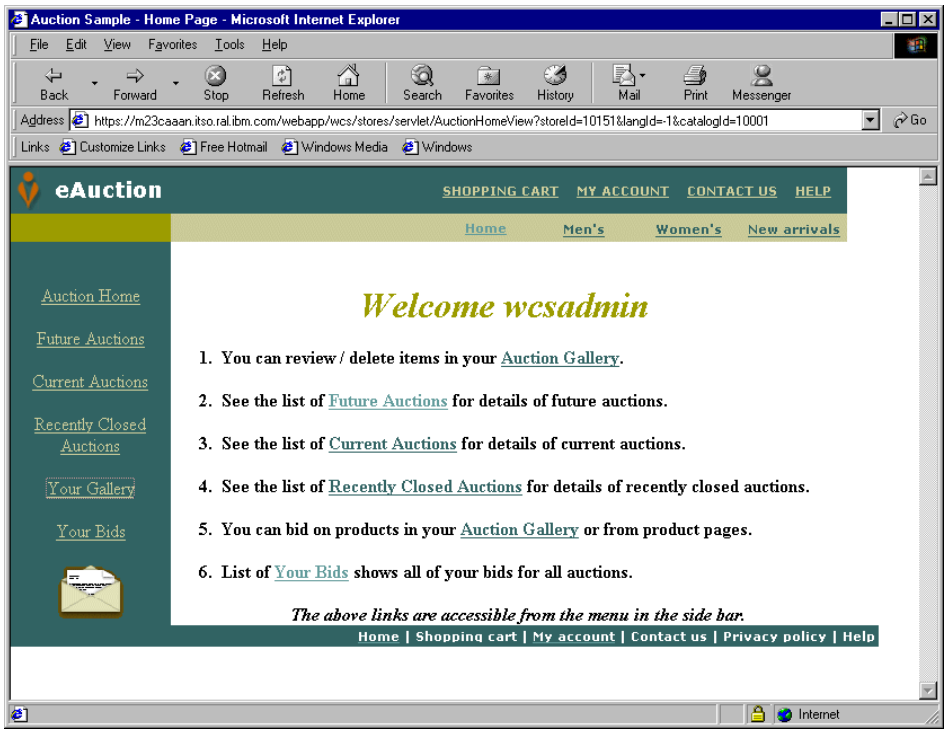


Figure 14-6 eAuction store home page - after changes (no frames)

In order for the Auction store to look similar, the frames that were used to display the navigation bar have been removed and the code to display the navigation bar has been added to the different JSPs that are accessed by commands in the auction pages. The eAuction sample store view commands did not require the store ID, catalog ID and language ID to be passed as name value pairs in the URL. But in order to include the header and footer and to make the links within active, the above-mentioned parameters are required in the URL. The links in the navigation bar have also been modified to include these parameters.

The list of view commands affected and corresponding JSPs that were modified are listed Table 14-1.

Table 14-1 Modified auction JSPs

Auction view	JSP file
AuctionHomeView	auc_main_home.jsp
DisplayHomeView	auc_home.jsp
AuctionListView	auc_all_auction_list.jsp

Auction view	JSP file
MailListView	auc_maillist.jsp
ShopperBidListView	auc_shopper_bid_list.jsp
ShopperAuctionListView	auc_shopper_auction_list.jsp

14.2.3 Enable an existing store for auctions

In this section we will discuss the steps involved in adding the auction feature to any existing store.

1. Decide which page on your existing store will provide the link to auctions. Make code changes in the JSP, similar to the changes made to the account.jsp file and the infashion.properties file in the example.
2. The modifications made to the JSPs as listed in Table 14-1, have used include files. For example the header and footer JSPs are include files within the auction pages. Therefore, as long as you have a header.jsp and footer.jsp in your existing store you should be able to just copy the auction JSPs to your store and have it working without a problem.
3. The auction JSPs have been modified to include code for the side navigation bar. The modified JSPs that are being provided for download with this book have a very clearly commented code section for the start and end of the side navigation bar. You might have to modify this section of the code as applicable to your environment.
4. The auc_common.jsp should also be modified to ensure the auction pages can access the store name and the resource/properties files used by the store. Refer to the previous section for code example for auc_common.jsp.

14.3 Create an auction

Auction creation involves checking inventory, updating inventory, obtaining and saving fulfillment center information, checking and validating for bid rules and finally inserting into the auction table and updating tables related to the above mentioned tasks.

WCS V5.1 provides a couple of methods to create auctions. One is the simple and easy-to-use GUI interface in the Commerce Suite Accelerator and the second is a command interface that could be invoked either manually or through a customized GUI that is part of a store. In this section we will investigate the steps involved in using both these methods for auction creation.

14.3.1 Auction wizard

In order to create an auction using the Commerce Suite Accelerator, you need access to the Merchandising menu. Merchandising Manager, Marketing Manager and Merchant are the Access groups that can access the Merchandising menu.

Create a sample auction

1. Log on to Commerce Suite Accelerator as a user with at least one of the above-mentioned access levels.
2. Select the **Merchandising** menu and then choose **Auctions**.
3. The next page provides two options: to create a new auction or to find an existing auction. Click **New**.
4. The Auction types page lets you select the standard types (see 14.1.2, “Types of auctions” on page 402) or customized auction styles (see 14.1.4, “Auction style” on page 404). For this example we will select **Create New -> Open Cry Auction**.
5. In the auction Product page, select **Find** to search for a product by name. Entering `Cords` in the Name field and clicking **Find** would give a results window like the one shown in Figure 14-7. This is assuming you have created a store using the sample Web Fashion or InFashion store archive.

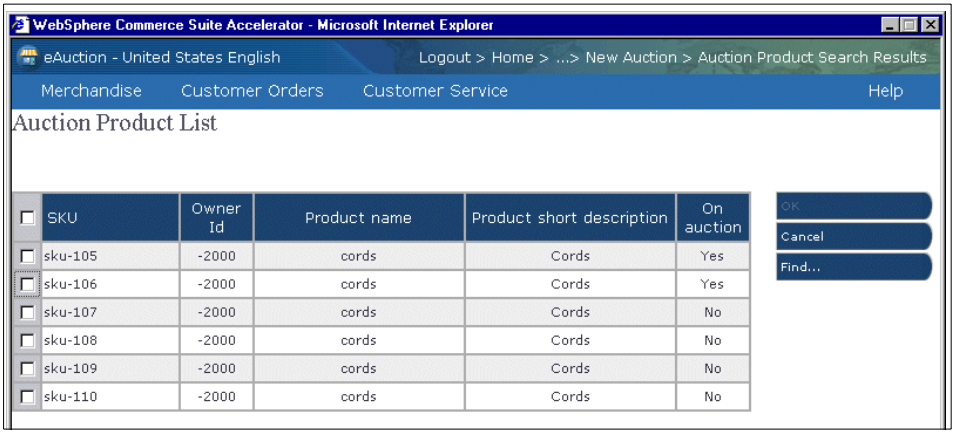


Figure 14-7 Product search result - cords

Note: Although check boxes are available for each product row, selecting more than one check box disables the OK button. Therefore you will be able to select only one product at a time to create an auction.

6. Select **sku-107** and click **OK**.
7. This takes you to a page that allows you to enter the quantity that you would like to put on auction. The validation for available quantity is not done at this point. Therefore it would be wise to confirm the availability of the quantity you wish to auction by querying the Inventory table. You can do this either manually or by using Commerce Suite Accelerator's GUI interface. From the the Merchandising menu, select **Products**. This displays all the available products. Select the desired product and click **Inventory**. The resulting page looks like Figure 14-8.

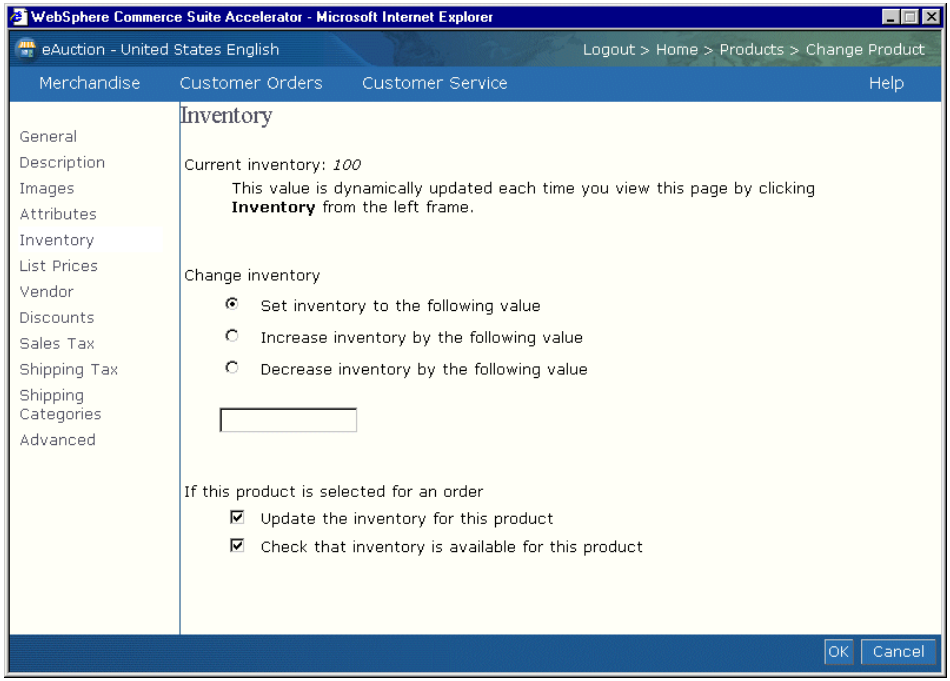


Figure 14-8 Inventory for product with SKU sku-107

8. Enter a quantity of 1000 in order to test the scenario of not being aware of the available quantity before the start of the auction creation process.
9. The next couple of pages allow you to enter the desired duration, price and pricing mechanism.

   Enter a valid date and time for the start and end of the auction. You can also set a rule for closing the auction based on the number of days with no bids.
10. The Auction display page allows you to specify what templates to use in displaying auction rules and the product display for an auction. It also allows you to enter the desired product description for display.

We will leave the default values auc_rule.jsp and auc_ItemDisplay.jsp for the Auction Rule template and the Product Display Template respectively.

11. The last page before completion of an auction creation is the Bid Control Rule page. If you have already created and saved a bid rule, it can be assigned to the auction. Refer to 14.1.3, “Auction rules” on page 403.

12. Clicking **Finish** will result in the following error message:



Figure 14-9 Invalid quantity error message

The auction creation pages do not provide a way to find the available quantity. As indicated earlier, find the available quantity by querying the inventory and catentry tables. For example the following query will return the available quantity for a product with the name “cords”, partnumber sku-107 and in the store with an ID of 10051.

```
select i.quantity from catentdesc c, inventory i, catentry ce where
ce.partnumber like '%sku-107%' and c.name like '%cords%' and c.catentry_id
= ce.catentry_id and ce.catentry_id = i.catentry_id and i.store_id = 10051
```

13. Navigate back to the Product page by clicking **Previous**, change the quantity to 100 or less and then navigate back to the final page and click finish. This will return the following message indicating successful creation of the auction.

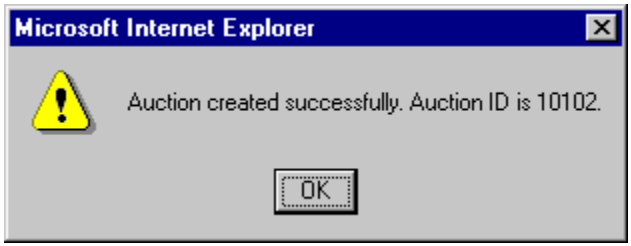


Figure 14-10 Auction created successfully

14.3.2 CreateAuction command

Commerce Suite Accelerator involves the **`CreateAuction`** command with the required parameters, which in turn creates an auction by completing the following tasks:

- Validate availability of product for purchase
- Validate for duplicate auctions
- Check inventory
- Update inventory
- Insert into auction table
- Update the onauction column in the catentry table
- Update the long and short descriptions in the auctdesc table

Refer to Figure 14-11 for the **`CreateAuction`** command syntax.

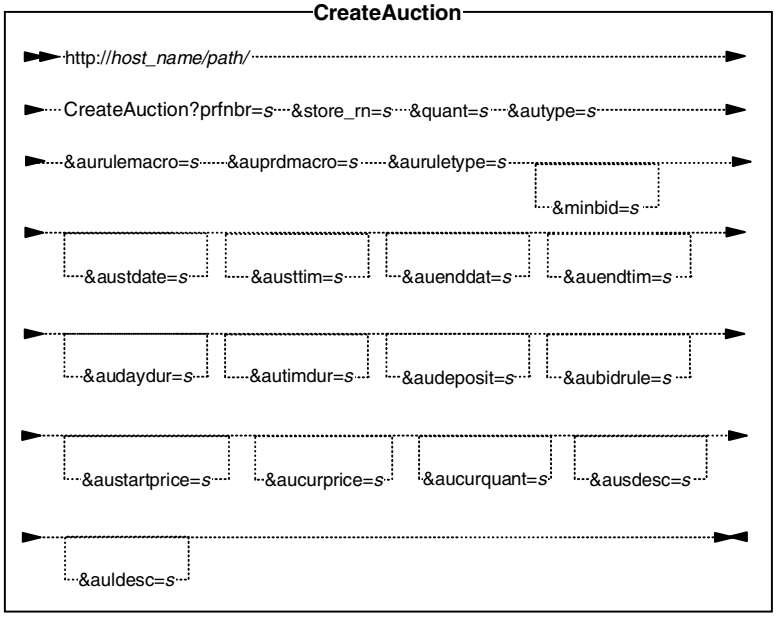


Figure 14-11 CreateAuction command syntax

14.4 Auction database tables

This section describes the auction database tables.

14.4.1 Auction tables

Auctions that have closed, though their status has changed, still have an entry in the auction table. To delete such obsolete auction records, do the following:

1. Ensure that WCS_HOME\wcs\bin and DB2_HOME\sqllib\bin are in the PATH environment variable.
2. Change to the directory to which you want log files written.
3. Type the following:

   ```
   dbclean -table auction -type <typename> -db <database> -days <daysold>
   -loglevel <loglevel>
   ```

   For the `-type` parameter, you can specify `settlement_closed` to indicate a completed auction record, or `retracted` to indicate a retracted auction.
4. Examine the dbclean.log_yyyy.mm.dd_hh.mm.ss.zzz file to verify that the command was successful.

Follow the above steps to clean up other auction tables such as auctionlog, autobidlog and bidlog.

14.4.2 CleanJob

Creating and managing auctions is dynamic, highly time sensitive and involves a number of tasks to be executed in the background.

These background tasks are invoked by the Job Scheduler. Refer to 14.1.6, “Scheduled jobs” on page 404. The schstatus table is updated each time the Job Scheduler executes a command. Therefore, over time the schstatus table can get overloaded with data. In order to keep the Job Scheduler performing efficiently, it is important to clean up the tables that are frequently used by it. The following is a sample command for cleaning up the schstatus table of the obsolete auction records:

```
http://myhostname/webapp/wcs/stores/servlet/CleanJob?endTime=2001:10:05:15:29:0
6&URL=basemall.JSP
```

The above command will delete all job instance entries in the schstatus table that were completed before the time specified, which is 2001:10:05:15:29:06.

Refer to Figure 14-12 for CleanJob command syntax.

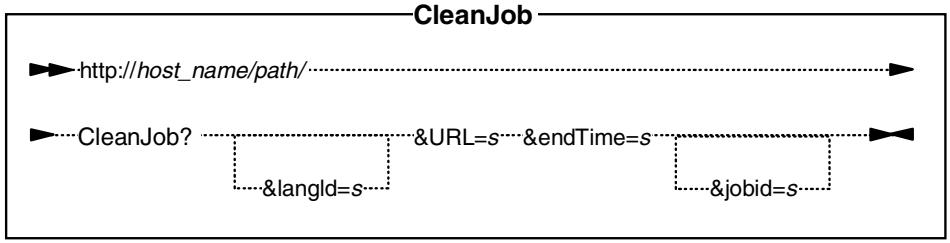


Figure 14-12 CleanJob command syntax

14.4.3 Auctions and bidding data model

The tables Auction, Bid, Auto Bid, Catalog Entry, Store, and Member, which enable auctions and bidding in WCS, are shown in Figure 14-13.

- Auctions are created by merchants (Members).
- Each auction sells only one type of product (Catalog Entry).
- Customers place bids (Bid) against one or more auctions. Customers may also set up autobids (Autobid), which automatically generate bids until the customer wins the bidding or a maximum bid price is reached.
- A deposit (Bidpayment) may be required of bidders.
- Merchants can create auction styles (Auctstyle), which are used as templates to simplify auction creation.

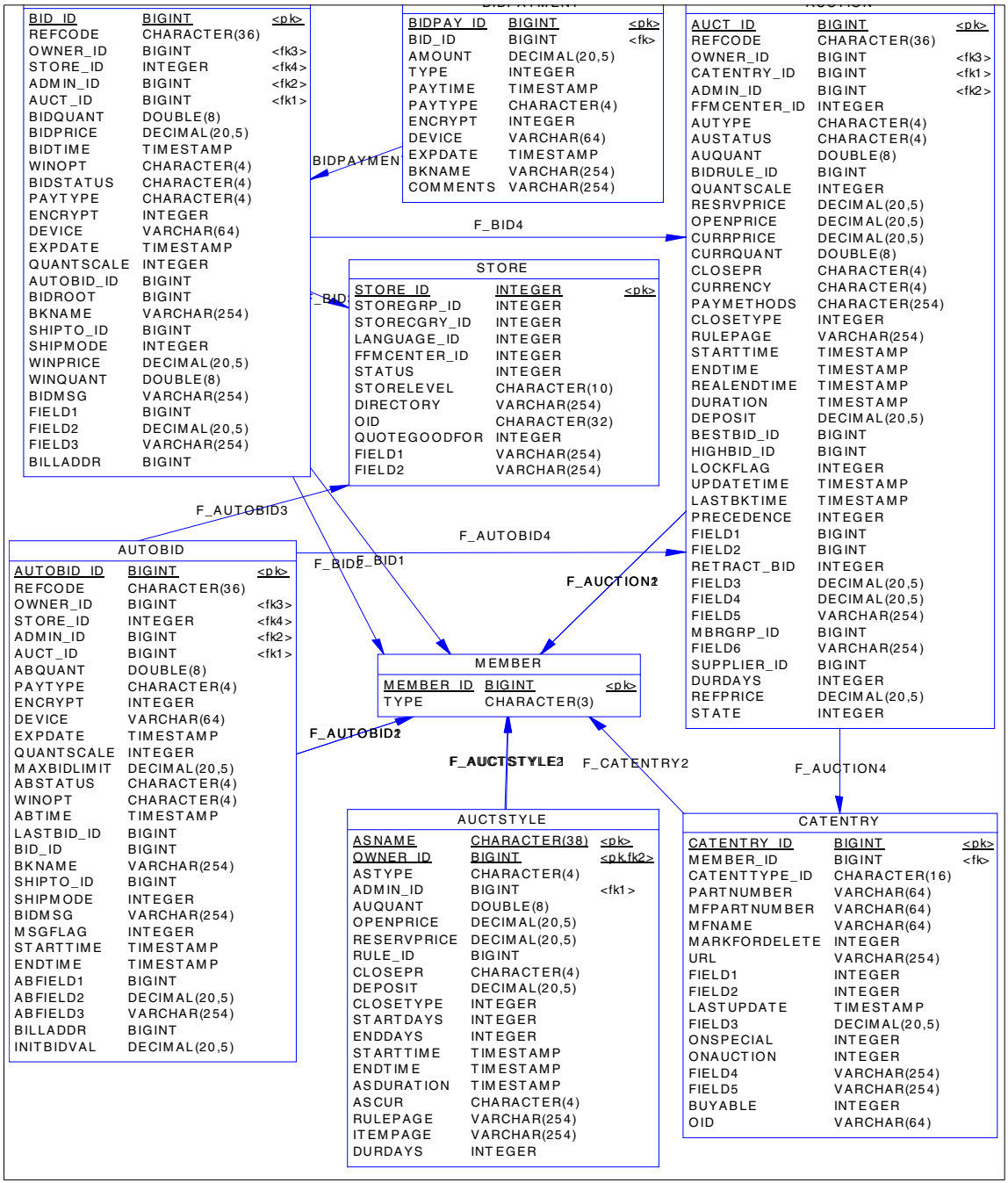


Figure 14-13 Auction bidding data model

Part 4

Managing a store

© Copyright IBM Corp. 2001

15

Deployment and catalog data management

After the initial deployment of the store and database assets, the catalog must be maintained and updated with new data. There are different ways to add, update and maintain data for a WebSphere Commerce Suite Web site. The key to determining the right method to use is determining the amount of data to load, the number of tables that will be impacted, the type of runtime environment, the available hardware infrastructure, and possibly other parameters that affect performance and system downtime.

This chapter describes the different ways to deploy a store and load data into the WCS database. In addition, we will provide an overview of the WCS V5.1 loader package, and demonstrate by example how to use the Loader package to add data to the WCS database.

This chapter is organized into the following sections:

- Deployment
- Loader package
- Catalog assets in a store archive
- Where to find more information

© Copyright IBM Corp. 2001

15.1 Deployment

Store deployment refers to the process of publishing your Web assets and database assets to the WCS runtime environment. In some cases you may publish a store archive including all store contents. For example, a new store or new release of your store may include Web assets, database contents, properties files, XML descriptor files. In other cases, you may want to just update product data.

The deployment of a WebSphere Commerce Suite store includes the file transfer of Web assets such as JSPs, HTML, images, JAR files containing Java classes, EJBs, and WCS database updates via the loader package. In addition, new Java assets require configuration updates to the WebSphere Commerce Server - <wcs_instance> command line arguments.

15.1.1 Deployment runtime environments

During the development cycle there may be several different runtime environments that a store is deployed to. Some common examples of runtime environments include the following:

- Developer runtime environment

  This WCS V5.1 runtime environment is installed on the developer's workstation or test server and is used to deploy the store and unit test code being developed. Variations of this type of runtime environment include several developers sharing this environment.

- Test runtime environment

  Builds of a store of the collective work of the development team are deployed to this runtime environment. System testers then verify functionality of the store being developed.

- Staging runtime environment

  When preparing to roll out a new or updated WCS store, many customers will deploy their store to a staging environment to work any kinks out of the system before going live in the production environment. This type of environment can avoid costly downtime.

- Production runtime environment

  This runtime environment will host the deployed store for customers to access your commerce Web site to browse and purchase goods and services.

15.1.2 Methods of store deployment

In this section we provide a description of the tools and processes used for store deployment.

Store Services store deployment

Store Services is a Web browser-based GUI that enables the user to select a Store Archive File (SAR) to be published. A SAR file is a compressed archive that encapsulates all the components of a store.

Throughout this redbook, we have included several examples of this type of deployment. Refer to 6.7, “Deploy the sample store” on page 115 for detailed information.

Command line store deployment

SAR files publishing can also be initiated via the command line using a script file supplied with WCS.

Refer to 7.7.2, “Deploy the sample store via the command line” on page 175 for detailed information.

WebSphere Commerce Studio

WebSphere Commerce Studio is a Web design and development workbench that enables graphical editing of Web pages and visual assembly of components such as JSPs and static HTML. It integrates with IBM VisualAge for Java to form a complete development environment.

Studio contains functionality to file transfer store assets to the runtime environment via FTP, network file system, and local file system. This type of deployment is more efficient when in the development stage. In addition, Studio can be configured to publish the assets to the local file system in a series of steps for creating your own SAR file for deployment.

Refer to 11.5, “Publishing from Studio” on page 346 for detailed information.

File transfer and third-party tools

Individual files can be transferred to the production system directories using FTP or another file transfer mechanism, provided the user is authorized. This method of deployment should be considered carefully to avoid the possibility of damaging the integrity of the system should a mistake occur.

15.2 Loader package

The WCS loader package consists of utilities for preparing and loading large amounts of data into the WCS database.

The components of the loader package are:

- DTD Generator

  The DTD Generator creates a DTD and schema to be used by the loader package. The DTD Generator uses an input file containing the WCS database table names. Depending on how the **`DTD Generate`** command is invoked, it generates either a Document Type Definition (DTD) or a DTD and a schema with a detailed XML file describing the database.

- ID resolver

  The ID Resolver generates identifiers for XML elements that require them. If your XML content already supplies identifiers, you do not have to run the ID Resolver. These are examples of situations in which you might want to use the ID Resolver:

  – When you are loading new content in XML format, and identifiers for the data are required.

  – When you are updating content where identifiers already exist for an object in the database.

- Loader utility

  The loader utility loads valid and well-formed XML input into the database. This tool is similar to the Mass Import utilities within past versions of WCS. Data must be in XML format with an associated DTD. The **`Load`** command is used to load data using this utility.

15.2.1 Loader package data load process

The loading process consists of steps necessary to move data into your Commerce Suite database, including the following:

1. Determine the subsystem, database schema and in particular the tables to load data into.
2. Generate a document type definition (DTD) using the DTD Generator utility. The DTD Generator analyzes table definitions in the database to discover the relationship between tables and primary key and indexed columns. Create data input files in XML format.
3. The ID Resolver substitutes symbols with unique values determined from database primary keys or unique indexes. If a unique index does not exist on a table, additional hints can be provided in IDResolveKeys.properties.

4. Loading the data using the loader utility.

Figure 15-1 illustrates the data load process using the loader package.

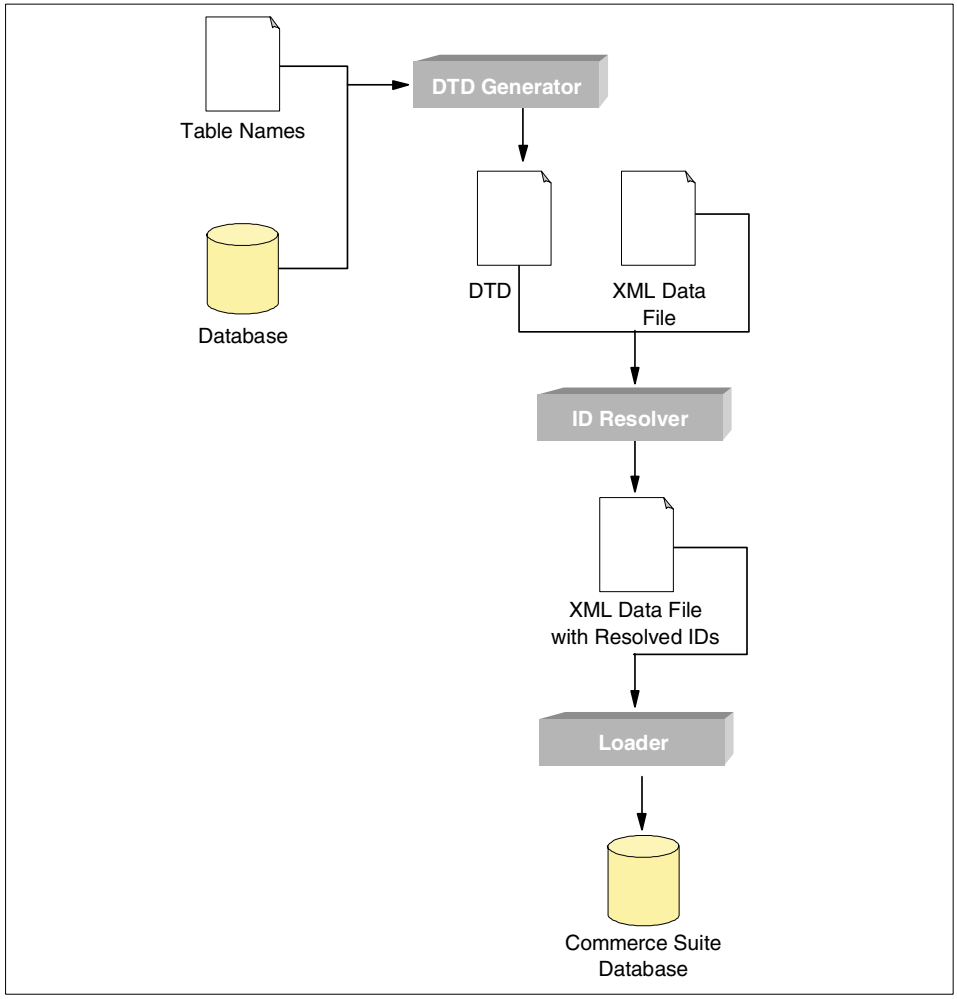


Figure 15-1 Loader package data load process

Important: We strongly recommend that you back up the WCS database before using the loader utility. In the event that you have errors during the execution of the load, it is easier to roll back from a backup database rather than fix the errors.

15.2.2 Sample data load using the loader package

This section explains in detail the steps required for using the WCS loader package. We will use sample data to load into the CATEGORY table.

To load data using the loader package, complete the following steps:

1. Update the system classpath with the following paths (for example, on Windows NT or Windows 2000):

```
c:\ibm\wcs\lib\loader\DTDGenerator.zip
c:\ibm\wcs\lib\loader\Logger.zip
c:\ibm\wcs\xml\loader
c:\ibm\wcs\lib\loader\IdResGen.zip
c:\ibm\wcs\lib\loader\MassLoader.zip
c:\ibm\wcs\lib\loader\WCALogger.zip
c:\ibm\wcs\lib\loader\PropGen.zip
c:\ibm\wcs\lib\loader\SAFServ.zip
c:\ibm\wcs\lib\loader\WCMCommon.zip
c:\ibm\wcs\lib\loader\WCSTasks.zip
c:\ibm\wcs\lib\loader\wcmxmlp.jar
c:\ibm\wcs\lib\jlog.jar
c:\ibm\wcs\lib\loader\db2\dbconnect.zip
c:\ibm\wcs\lib\loader\jgl2.0.0.jar
```

   Where c:\ibm\wcs is the WebSphere Commerce Suite install path.

   Note: Each of the paths and files names need to be separated by a semicolon and should be typed in without spaces.

   On AIX or Solaris, replace c:\ibm\wcs with the appropriate path.

2. Create a file called tablenames.txt. This file needs to include all tables that you intend loading data into. The table names should be separated by a carriage return. For example, we will load data into the following tables:
   - CATALOG
   - CATALOGDSC
   - CATGROUP
   - CATGRPDESC
   - CATTOGRP
   - CATGRPREL
   - CATENTRY
   - CATENTDESC
   - CATGPENREL
   - CATENTREL
3. Create the DTD for the tables listed in the tablenames.txt file by executing the **`DTD Generate`** command from the command line.

For example:

```
> java com.ibm.wca.DTDGenerator.GenerateDTD -dbname eauction -dbuser
db2inst1 -dbpwd db2inst1 -outfile catalog.dtd -infile tablenames.txt
```

The syntax for the **DTD Generate** command is displayed in Figure 15-2.

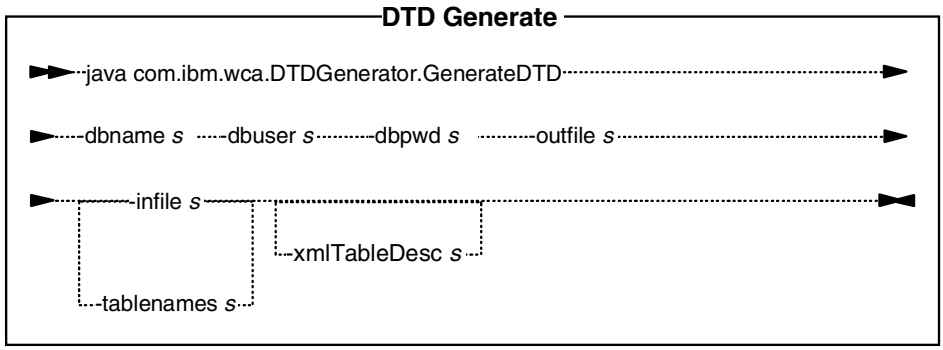


Figure 15-2 DTD Generator syntax

4. Review the output file (for example, catalog.dtd) created by the **DTD Generate** command (input file tablename.txt).

Example 15-1 Output file of DTD Generate command (catalog.dtd)

```
<!ELEMENT MALL (( CATALOG | CATALOGDSC | CATGROUP | CATGRPDESC | CATTOGRP |
CATGRPREL | CATENTRY | CATENTDESC | CATGPENREL | CATENTREL)*)>
<!ELEMENT CATALOG EMPTY>
<!ATTLIST CATALOG
CATALOG_ID CDATA #REQUIRED
MEMBER_ID CDATA #REQUIRED
IDENTIFIER CDATA #REQUIRED
DESCRIPTION CDATA #IMPLIED>

<!ELEMENT CATALOGDSC EMPTY>
<!ATTLIST CATALOGDSC
CATALOG_ID CDATA #REQUIRED
LANGUAGE_ID CDATA #REQUIRED
NAME CDATA #REQUIRED
SHORTDESCRIPTION CDATA #IMPLIED
LONGDESCRIPTION CDATA #IMPLIED
THUMBNAIL CDATA #IMPLIED
FULLIMAGE CDATA #IMPLIED>
```

5. Create the XML data input file that conforms to the DTDs that were generated. For guidelines on creating XML data input files, refer to WCS online help (search on create catalog assets). Example 15-2 displays a sample data input file.

Example 15-2 Sample data input file

```
<!DOCTYPE MALL SYSTEM "catalog.dtd">
      <MALL>
      <catalog
      catalog_id="@catalog_id_1"
      member_id="-2000"
      identifier="eAucOnline"
      description="eAuction site"
      />
      <catalogdsc
      catalog_id="@catalog_id_1"
      fullimage="webbapp\images\InFashion.gif"
      language_id="-1"
      longdescription="The eAuction site is a integration of the NetFashion ad
      Sample Auction stores"
      name="eAuc1"
      shortdescription="eAuction site"
      thumbnail="InFashion.gif"
      />......
```

6. Create an ID resolved version of the XML data input file. For example, type the following to execute the **ID Resolve** command:

   ```
   > java com.ibm.wca.IdResGen.IdResolve -dbname eAuction -dbuser db2inst1
   -dbpwd db2inst1 -infile catalog.xml -outfile catalogResolved.xml -method
   load
   ```

   The **ID Resolve** command syntax is displayed in Example 15-3.

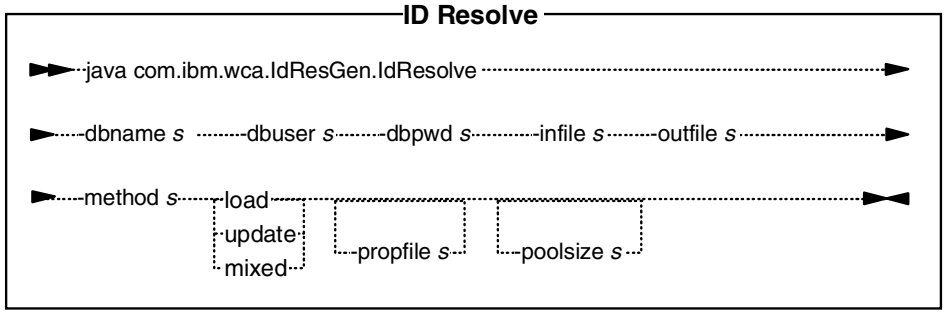


Figure 15-3 ID Resolve command syntax

Example 15-2 displays a sample ID resolved data input file. This is the output of the **ID Resolve** command.

Example 15-3 Resolved data input file

```
<MALL>
<CATALOG
```

```
CATALOG_ID="10051"
MEMBER_ID="-2000"
IDENTIFIER="eAucOnline"
DESCRIPTION="eAuction site"
/>
<CATALOGDSC
CATALOG_ID="10051"
FULLIMAGE="webbapp\images\InFashion.gif"
LANGUAGE_ID="-1"
LONGDESCRIPTION="The eAuction site is a integration of the NteFashion ad Sample
Auction stores"
NAME="eAuc1"
SHORTDESCRIPTION="eAuction site"
THUMBNAIL="InFashion.gif"
/>
....
....
</MALL>
```

7. Invoke the **Load** command to load the ID resolved XML input file. For example, type the following to execute the **Load** command:

   ```
   > java com.ibm.wca.MassLoader.MassLoad -dbname eAuction -dbuser db2inst1
   -dbpwd db2inst1 -infile catalogResolved.xml -method load
   ```

   The **Load** command syntax is displayed in Figure 15-4.

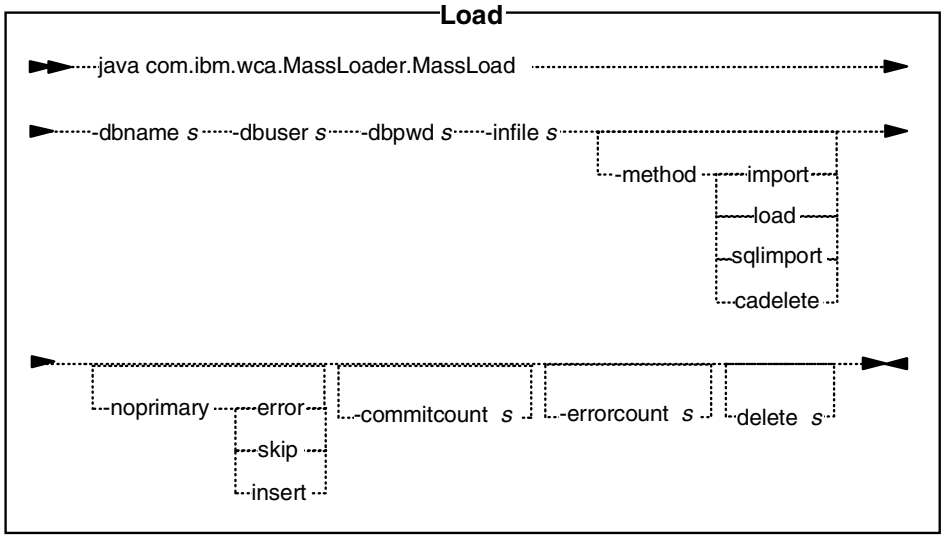


Figure 15-4 Load command syntax

Note: If you execute the loader utility from a remote machine (other than the DB2 machine), the load option is not applicable for the method parameter in the **`ID Resolver`** and **`Load`** commands.

Tables that include referential or check constraints are placed in check pending state. Summary tables that are defined with REFRESH IMMEDIATE, and that are dependent on tables being loaded, are also placed in check pending state. Issue the SET INTEGRITY statement to take the tables out of check pending state.

15.2.3 Loader package usage considerations

There are several issues to consider when using the loader package:

- Location to run the utility
- Size of the data load
- Length of required downtime
- Network traffic
- Sensitivity of data

If large amounts of data are to be loaded, it will obviously take more time, which impacts the downtime. The load on the network and the sensitivity of data might require that the loader be run from the database machine.

In order to run loader package from the database machine or another remote machine, the loader package and supporting files must be packaged. Refer to 7.8.10, “Step 10: Copy the loader package” on page 189 for details.

15.3 Catalog assets in a store archive

The store archive file is a packaged version of all the assets that make up a store. At a high level, the archive file can be classified into Web assets, properties files, descriptor, and database assets. Within this chapter, we are interested in the database assets, which are the DTD files and XML files that contain the database schema, store, catalog, and other data.

If you create your store archive using the tools in Store Services, your new store archive will initially contain the same store database assets as the sample store archive on which you based it (for example, webfashion.sar). The store database assets take the form of XML files.

In this section we will discuss the data assets in the store archive and look at different ways to update the data assets.

The sarinfo.xml contains information on each of the data assets. To determine which XML file to update for a particular subsystem, look within the asset element (tag) in the sarinfo file, with the name of the subsystem you wish to update. For example, if you wish to update the catalog subsystem, the section of the sarinfo file shown in Example 15-4 provides the list of XML files to update.

Example 15-4 Sample sarinfo.xml file

```
<asset
   fragmented="yes"
   name="catalog">

   <file
      name="data/catalog.dtd"
      type="dtd"/>
   </file>

   <file
      name="data/catalog.xml"
      priority="2"
      type="db-load"/>
   </file>
   <file
      name="data/en_US/catalog.xml"
      priority="3"
      type="db-load">
      <locale>en_US</locale>
   </file>

   <file
      name="data/es_ES/catalog.xml"
      priority="3"
      type="db-load">
      <locale>es_ES</locale>
   </file>
</asset>
```

As seen in Example 15-4, the catalog.dtd and the catalog XML files make up the catalog database asset. To change or add catalog data, modify the catalog.xml file in the locale that is applicable to your implementation.

Once changes have been made to the XML files, there are two options to load the data into the database:

1. Add the XML files back into the SAR file and re-publish the store. Refer to Chapter 12, “Creating a store archive (SAR) for deployment” on page 355 for

details. During the publishing of a SAR from Store Services, the loader package is invoked.

2. Loader package data load.

   The modified XML files could be used as input to the loader utility after its identifiers have been resolved. Refer to 15.2.2, “Sample data load using the loader package” on page 430 for more details on using the loader package.

15.4 Where to find more information

- WebSphere Commerce Suite V5.1 online help
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174

16

Administrative tools, tasks, logs, and troubleshooting

Once the e-commerce Web site has been deployed there are many activities that need to be performed to manage the site. In this chapter, we highlight some of the administrative tasks and utilities included in WebSphere Commerce Suite V5.1.

This chapter is organized into the following sections:

- Administration tools overview
- Caching
- The dbclean utility
- Job Scheduler utility
- Logging and troubleshooting

© Copyright IBM Corp. 2001

16.1 Administration tools overview

WebSphere Commerce Suite V5.1 includes several tools used for configuration, deployment, administration, business operations, and maintenance. In this section, we provide an overview of the following tools included in WebSphere Commerce Suite V5.1:

- WCS documentation
- WCS Configuration Manager
- WCS Store Services
- WCS Administration Console
- WCS Accelerator
- WebSphere Commerce Analyzer
- HTTP Server Administration Console
- WAS Administration Console
- DB2 Control Center

16.1.1 WCS documentation

Along with the many PDFs available on the WebSphere Commerce Suite V5.1 product CD, WCS V5.1 provides an extensive online help repository of information. We found that the WCS online help documents many topics that are not included in the PDFs.

Most noteworthy in the WCS online help (for this section of the book) are the instructions for administrative tasks based on the role of the I/T specialist or business user. The WCS online help includes task information for the following roles:

- Site administrator
- Store developer
- Marketing and merchandising manager
- Order clerk
- Customer service representative

To access on the online help on a Windows NT or Windows 2000 system, click **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Documentation**. The search facility within the online help can be used to find information on the administrative tasks.

16.1.2 WCS Configuration Manager

The WCS Configuration Manager is a Java application-based tool used to create and configure a WCS instance. The Configuration Manager (client) has a corresponding server, called the Configuration Manager Server, which must be started to use the Configuration Manager.

The Configuration Manager provides configuration tabs to enter information on instance, WebSphere Application Server, Web Server, Payment Manager, messaging, and session management. The information entered in the Configuration Manager is stored in the WCS instance XML configuration file and the WAS repository database.

For detailed information on creating a WCS instance using the Configuration Manager, refer to the following, depending on your operating system platform:

- Windows: refer to 6.6.1, “Creating a WCS instance using the Configuration Manager” on page 111.
- AIX: refer to 7.6.1, “Creating a WCS instance using the Configuration Manager” on page 166.
- Solaris: refer to 8.7.2, “Creating a WCS instance using the Configuration Manager” on page 259.

16.1.3 WCS Store Services

The WCS Store Services Web browser-accessible tool allows you to quickly create a store archive based on a sample store. Once the store archive has been created, Store Services can be used to perform the following tasks:

- Publish (deploy) the store to the WCS runtime environment.
- Change tax settings using the Tax notebook.
- Change shipping settings using the Shipping notebook.
- Change general store information using the Store Profile notebook.

Once a WCS instance has been created, Store Services may be used. To start the WCS Store Services, enter the following URL in the Internet Explorer V5.5 browser:

```
http://<hostname>/storeservices
```

For detailed information on using Store Services, refer to 6.7.1, “Publish store from Store Services” on page 115.

16.1.4 WCS Administration Console

The WCS Administration Console is a Web browser-accessible tool that allows you to maintain your online stores by completing administrative operations. When you log on to the Administrator Console, you select the store and language with which you want to work, if you are authorized to work with multiple stores and languages. The tasks that you are authorized to perform are displayed on the Administration Console home page through various menus. These tasks are based on the user roles and authority levels.

The following list outlines what tasks can be performed by each role:

- Site Administrator
  - Manage administrators and access groups
  - Restrict customer commands from stores
  - Define transports and message types for the site
  - Monitor performance of the site
  - Specify Payment Manager settings
- Store Administrator
  - Configure messages
  - Administer Blaze rules

This tool is used after the store has been published from Store Services. To start Store Services, enter the following URL in the Internet Explorer V5.5 browser:

```
http://<hostname>/adminconsole
```

16.1.5 WCS Accelerator

The WCS Accelerator is a Web browser-accessible tool that allows you to maintain your online stores by completing various store operations. If you are authorized to work with multiple stores, you select the store and language with which you want to work when you log on to the Commerce Suite Accelerator. If you are authorized to work with a single store, the store name is pre-selected during logon, and if the store supports more than one language, you select the language with which you want to work.

Tasks that you are authorized to perform in your role are displayed on the Commerce Suite Accelerator home page menus. These tasks are based on access groups and authority levels, which are defined by the Site Administrator by using the Administration Console. Table 16-1 outlines the available menus in the Accelerator (WebSphere Commerce Suite V5.1 Pro Edition), who has access to the menus, and what tasks can be performed using the menus.

Table 16-1 WCS Accelerator tasks based on access group

Menu name	**Access group**	**Task**
Store	► Merchant ► Marketing mgr ► Merchandising mgr	Manage business intelligence reports
Marketing	► Marketing mgr ► Merchandising mgr ► Merchant	Manage customer profiles: ► List customer profiles ► Create, change, delete, and duplicate profiles ► View a summary of each profile Manage campaigns: ► List campaigns ► Create, change, and delete campaigns ► Publish campaigns ► List campaign initiatives Manage campaign initiatives: ► Create, change, and delete campaigns initiatives ► Publish campaigns ► View campaigns incentive statistics ► Create, change, and remove campaigns incentive conditions

Menu name	Access group	Task
Merchandise	► Merchandising mgr ► Merchant	Manage products: ► Find and list products ► Change product details ► List offers assigned to products ► Change offer details
		► Manage auctions: ► Find and list auctions ► Create, change, and retract auctions ► Close the bidding for auctions ► List and withdraw bids ► List discussion forums ► Create and delete discussion messages ► Respond to discussion messages ► Make discussion messages public ► List bid control rules ► Create, change, and delete bid rules ► List auction styles ► Create, change, and delete auction styles
		Manage discounts: ► List discounts ► Create and delete discounts ► View a summary of each discount ► Activate and deactivate discounts ► View Product Advisor statistics
Customer Orders	► Order clerk ► Merchant	Fulfill and process customer orders: ► Find and list customer orders ► View a summary of each order ► Change the status of orders ► Process payment ► Add comments to orders ► Access Payment Manager

Menu name	Access group	Task
Customer Service	► Customer service representative ► Merchant	Manage customer information: ► Find and list customers ► Change customer registration information, including passwords
		Manage orders for customers: Find and list orders for customers Place orders for customers View a customer's order history View a summary of each order Change order details Add comments to orders Cancel orders
		Manage auctions for customers: Find and list auctions Manage discussion forums, such as creating and responding to discussion messages, and making messages public Withdraw bids for customers

The WCS Accelerator can be started by entering the following URL in a Microsoft Internet Explorer V5.5 Web browser:

```
http://<hostname>/accelerator
```

Note: Once you have completed the tasks for a store, you should log out of the Commerce Suite Accelerator, rather than simply closing the browser. When you log out, your SSL cookie is dropped and you no longer have secure access to the Commerce Suite Accelerator. This is especially important if the Commerce Suite Accelerator will be used by multiple users on a single machine with different authorities. Logging out prevents unauthorized access.

16.1.6 WebSphere Commerce Analyzer

WebSphere Commerce Analyzer is an optional application included with the WebSphere Commerce Suite, V5.1 Pro Edition. If installed, it provides a robust business intelligence solution designed to analyze and report on the activities of your customers.

Note: WebSphere Commerce Analyzer is available only for the Window NT and Windows 2000 platform.

WebSphere Commerce Analyzer automatically extracts data from the production database on a regular basis, where it then processes logs and numerous database records to compile reports based on customer traffic and site usage. These reports, accessible from the Commerce Suite Accelerator, demonstrate comparable success rates of your marketing campaigns, as well as demographic distributions of your customers. These reports provide feedback that can be used to evaluate recent campaigns and to initiate change for upcoming campaigns. This completes the marketing campaign life cycle.

The data extraction schedule is fully configurable. A typical schedule would be to run the data extraction process on a daily basis in order to minimize the amount of data that is extracted during each run. WebSphere Commerce Analyzer is typically located on a remote dedicated machine to reduce any performance effects on the production machine.

When creating your next campaign it will be worthwhile to look at past campaigns in order to fine-tune your strategy. Commerce Analyzer can help you achieve this by providing feedback on:

- Campaigns and initiatives
  - What was displayed and clicked by customers
  - Did it lead to a sale or did the customer change his mind
- Sales and customers
  - Analyze sales by time period, geographic and demographic units
  - Look at what products were the most popular/least popular
  - Which products were looked at the most but not bought

The data for the above reports is extracted from the WCS database and parsed into the WebSphere Commerce Analyzer Datamart. Then it is processed and can be viewed from the Commerce Accelerator tool.

For more information about WebSphere Commerce Analyzer, refer to the IBM *WebSphere Commerce Analyzer User's Guide*, found on the WebSphere Commerce Analyzer product CD.

16.1.7 HTTP Server Administration Console

The IBM HTTP Server Administration Console is used to manage the IBM HTTP Server. The Administration Console is a Web browser-accessible tool, and requires the IBM HTTP Server Administration Server to be running to access it. The configuration settings modified by this tool are saved to the httpd.conf file.

To access the HTTP Server Administration Console, start the HTTP Administration Server, enter the following URL in a Web browser, and click **Configure server**:

```
http://<hostname>
```

16.1.8 WAS Administration Console

The WebSphere Application Server Administration Console is the primary configuration tool used to manage the WebSphere Application Server. The Administration Console requires that the Administration Server is running.

For detailed information on using the WebSphere Application Server Administration Console, refer to one of the following:

- For an example of configuring the WAS, refer to 6.4.4, “Configure the WebSphere Application Server” on page 106.
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161

16.1.9 DB2 Control Center

The DB2 Control Center is installed with the DB2 UDB V7.1, EE for Windows Administrative Client. The Windows-based GUI is very easy to use for managing DB2 databases on the local system or remote DB2 servers on various operating system platforms.

For more information on the DB2 Control Center, refer to the DB2 UDB V7.1, EE product documentation.

16.2 Caching

In this section we describe the different methods of caching available in WebSphere Commerce Suite V5.1, and explain how to configure and manage caching.

Caching is used to improve site performance for Web pages that are frequently accessed. For example, in a typical store 90% of shopper requests are for catalog pages. By caching the catalog display pages, we can invariable improve the speed for displaying these pages, and ultimately improve the Web site performance for the customer.

When a shopper tries to access a catalog page, most of the time is spent parsing the HTTP request, accessing the database, and dynamically creating the page the shopper wants to see. Therefore heavy site traffic and large amounts of catalog data would adversely impact caching. Since the catalog data does not change as often, it allows the flexibility of being able to serve an equivalent static page for those catalog pages that have not changed since it was last retrieved.

16.2.1 Types of caching

WCS V5.1 provides two ways to store static pages in a cache:

- Web server
- Commerce server

In order to use either of these caching methods, the command has to be cacheable. A command is cacheable when its HTML result does not vary for different users viewing the same command with the same parameters.

Web Server caching

Web server caching is performed by the WebSphere Application Server plug-in and only uses information in the URL for the cache search. Generally, you should enable Web Server caching unless you will use multiple currencies or member groups. By default the commands **`CategoryDisplay`**, **`ProductDisplay`**, **`TopCategoriesDisplay`** and **`StoreCatalogDisplay`** are enabled for caching.

Commerce Server caching

Commerce server caching is performed by the WCS server. This method of caching can be used for sites with distinct pages for member groups, multiple languages or multiple currencies.

16.2.2 Configure caching

The WCS Configuration Manager provides a GUI interface for configuring caching. The cache configuration settings are applied at the WCS instance level. Therefore, cache setting cannot be customized for individual stores within a WCS instance.

Cache wizard

The WCS Configuration Manager Cache wizard can be used to add a command to the cache.

To add a command to the cache using the Cache wizard, complete the following steps:

1. Start the Configuration Manager Server.
2. Start the Configuration Manager.
3. Select and expand your WCS instance.
4. Right-click **Caching Subsystem**, and select **Add a command to cache**.
5. When the Cache wizard appears, enter the cacheable URL and number of key sets (see Figure 16-1), and then click **Next**.

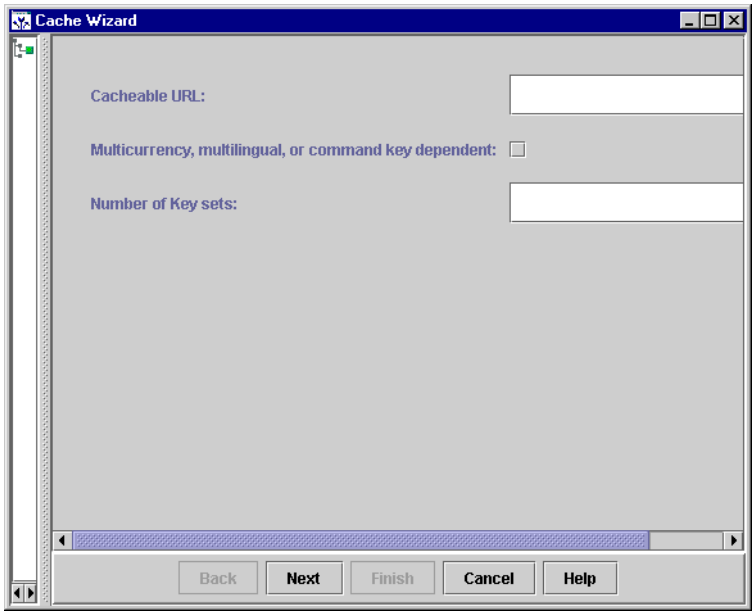


Figure 16-1 Cache wizard - cacheable URL

6. In the next Cache wizard window, enter the key set, hash key, member key, and number of keys (see Figure 16-2), and then click **Next**..

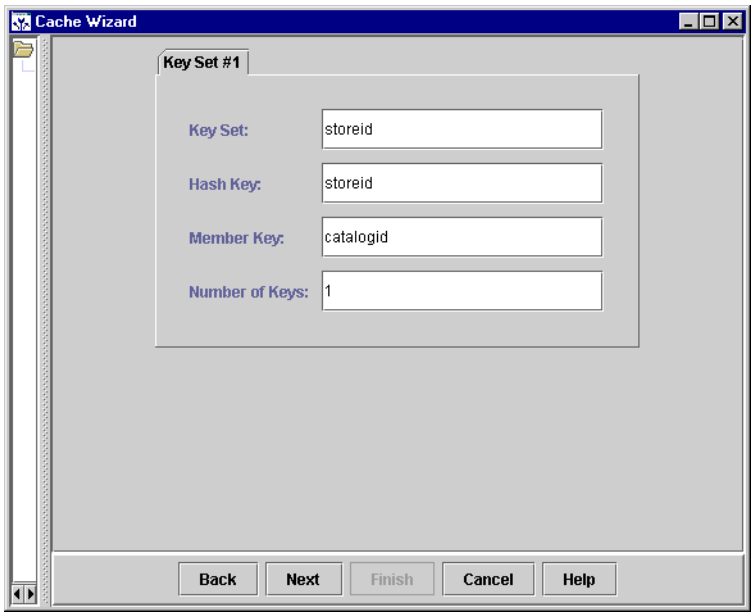


Figure 16-2 Cache wizard - key set information

7. In the next Cache wizard window, enter the key (see Figure 16-3) and then click **Next**.

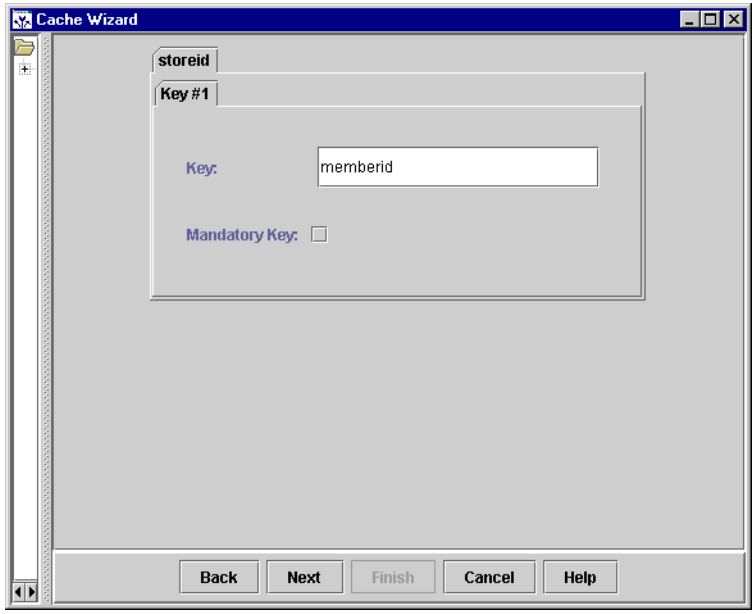


Figure 16-3 Cache wizard - key

8. Click **Finish** to save your changes to the WCS instance XML configuration file.

Caching parameters

Example 16-1 contains a sample of the default caching values set within the WCS instance XML configuration file found in the <wcs_install_path>\instances\<wcs_instance>\xml directory.

Although you could modify the WCS instance XML configuration file directly, we recommend that you use the Configuration Manager to configure caching.

Example 16-1 Default cache settings in the WCS instance XML configuration file

```
<Cache MaxObjectsPerMember="500"
         MaxAllowedRefreshPeriod="3600"
         CacheDirsPerMember="100"
         AutoPageInvalidation="true"
         CacheConnectionTimeout="120000"
         CacheCleanupAgentHostname="localhost"
         CacheDaemonMaxThreads="64"
         Enabled="true"
         useCacheCleanupTriggers="true"
         CacheFilePath="C:/ibm/WCS/instances/demomall/cache"
         CacheStoreClassName="com.ibm.commerce.cache.FileSystemCacheStore"
```

```
        CacheCleanupPollingInterval="600"
        CacheDaemonPort="16999"
        WCSAppPath="/webapp/wcs/stores/servlet"
        CacheCleanupAgentPort="80"
        CacheDaemonBindAddress="localhost">
    <CacheableURL sessionDependent="true"
                  name="StoreCatalogDisplay">
      <KeySet MemberKey="storeId"
              name="Key Set #1"
              HashKey="storeId" />
    </CacheableURL>
</Cache>
```

Cache optimization considerations

The following list provides some cache optimization considerations:

- If multilingual, multi-currency, member group-specific display pages are not required, ensure that the field multicurrency, multilingual, or command key dependent is not checked.
- Ensure directories that contain less than 1000 files are set with a value of the Cache directories per member field appropriately. For example, if you expect one store to have 50,000 files, set this to at least 50.
- If pages do not need to be automatically removed from the cache, ensure that the Auto Page Invalidation box is not checked. If any number of files can be in the cache, set the Max Objects Per Member field to 0. This disables tracking of objects in the cache.

16.2.3 Manage caching

This section includes some recommendations for managing caching.

- Cache invalidation triggers

  A trigger is a set of (SQL) statements that automatically “fires off" an action when a specific operation (such as changing data in a table) occurs. A trigger consists of an "event" (insert, delete, or update command) and a related "action" (insert, delete, update, or execute procedure command).

  The WCS cache uses triggers as a notification mechanism to indicate when an object is invalidated. To enable triggers, check **Enable triggers** on the **Caching** page of Configuration Manager. To remove triggers, uncheck **Enable triggers** then click **Apply**. You can enable and disable cache invalidation triggers by setting the “useCacheCleanupTriggers” property in the configuration file (wcs_instance.xml).

To cache URLs for your custom commands, you may also want to use the cache cleanup worker to ensure your pages are synchronized with the database. Create a trigger that will be invoked on insert, update or delete on the table that your command relies on. The trigger should insert a record into the CACHLOG table and set the CACHLOG.CACSTMP column to the current timestamp. If automatic page invalidation is enabled, the cache cleanup worker will remove the page.

Refer to WCS online help by searching for Cache Invalidation triggers to get a detailed table of the trigger group, action and tables affected.

- **`CacheDelete`** command

  The **`CacheDelete`** command allows you to initiate cache deletion without the need for access to the file system. It requires auto page invalidation to be enabled. This can be done either in the Configuration Manager or by setting the AutoPageInvalidation property to true in the XML configuration file (wcs_instance.xml).

  Note: AutoPageInvalidation enables the cache cleanup worker. This is required if you are going to use trigger-based page invalidation or the **`CacheDelete`** command. The default value is true.

  The command updates the CACHLOG table and the cache cleanup agent purges the corresponding files from the file system.

  ```
  http://hostname/webapp/wcs/stores/servlet/CacheDelete?storeId=10
  ```

  The above example URL would delete all cached pages for the store with a Store ID of 10. As indicated in the CacheDelete command syntax shown in Figure 16-4, you can also specify urlName, memberKeyName, hashKeyName or haskeyValue as additional deletion criteria.

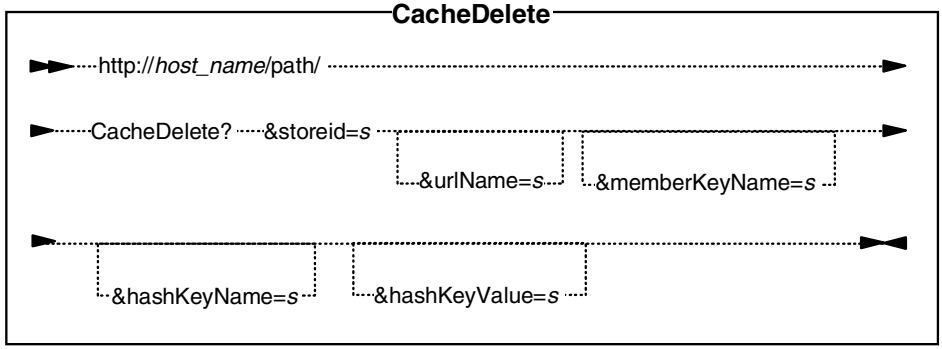


Figure 16-4 CacheDelete command syntax

16.3 The dbclean utility

The dbclean utility removes records from the WCS database tables. The utility queries the CLEANCONF table for information on the tables to be cleaned up and any conditional criteria to be followed. Any custom tables that require clean up could be added to the CLEANCONF table. Before adding any tables to the CLEANCONF table, it is important to understand the underlying database schema and the relationship with other tables. The cleanup utility obtains primary key and other referential integrity-related information for all the tables specified in the CLEANCONF table.

There are two delete modes, top-down and bottom-up. The default is top-down. In order to use this mode, you need to have a good understanding of the schema because it can fail if a delete-restrict is specified in the RI. The bottom-up method deletes all the child records before deleting the parent. Before using dbclean to remove records, it would be a good idea to use it with the check_table_only option. This will list all tables that may be impacted by your delete and all tables that have delete-restrict and thus helps you decide on the right delete mode for you.

If you set the loglevel to 2, this will trigger the bottom-up method of deletion, but will still fail if delete-restrict is specified on any of the tables. In order to force deletion, set the force option to yes.

16.3.1 Database tables and record types

Below shown is a list of tables, their corresponding record types, and descriptions that will be most commonly used with the database cleanup utility.

Table 16-2 dbclean tables

Table	Record type	Description
Member	organization	Deletes all organizations with the specified name.
Member	guest-shopper	Deletes all non-registered customers at least *n* days old with no orders. If the member has an order with a status of “Q”, it will prevent deletion.

Table	Record type	Description
Address	tempaddress_without_orderitems, obsolete	Deletes addresses with a status of “T”, at least *n* days old, and not referenced by any order items.
Orders	complete_order, cancel_order	Deletes all orders with a status completed order -“C” or canceled order - “X” not changed in the last *n* days.
Catentry	with_all_orderitems_iitems, without_orderitems, without_orderitems_iitems	Deletes all catalog entries marked for deletion and not changed in the last n days, and not referenced by the Auction table, and not referenced by the Orderitems table, or by the item table, respectively.
Auction	settlement_closed, retracted	Deletes rows not changed in the last n days, whose status is “SC” or “R” respectively.

For details on other tables specified in the CLEANCONF table, refer to the WCS online help and search for “about record types”.

16.3.2 The dbclean command syntax

The dbclean command syntax is shown in Figure 16-5.

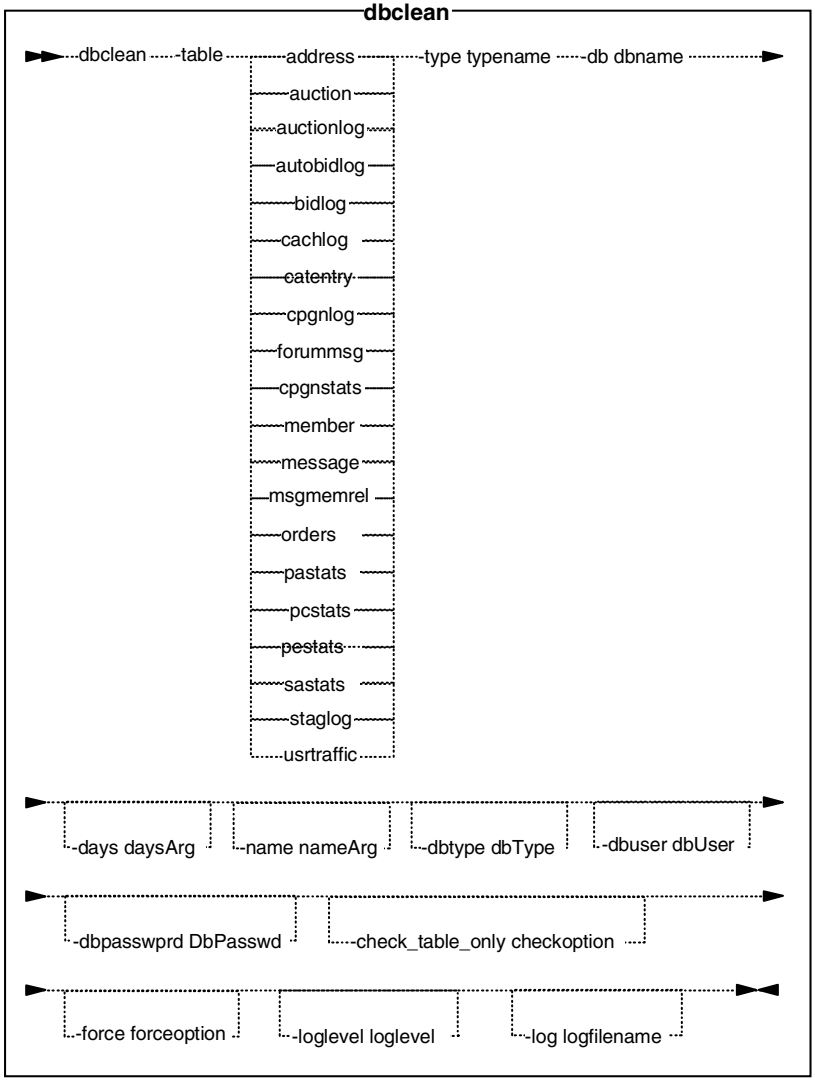


Figure 16-5 dbclean command syntax

Example 1

Below is an example of using the **dbclean** command to delete organizations with a specific name:

```
> dbclean -table member -type organization -db <database> -days <daysold> -name
<name of organization to delete> -loglevel <loglevel>
```

Example 2

Below is an example of using the **dbclean** command with the force option:

```
> dbclean -table <tablename> -type <typename> -db <database> -days <daysold>
-loglevel <loglevel> -force yes
```

Example 3

Below is an example of using the **dbclean** command to verify which tables specify the delete-restrict parameter:

```
> dbclean -table <tablename> -type <typename> -db <database> -check_table_only
yes
```

16.3.3 Using the dbclean command

To delete records using the **dbclean** command on the system where WCS is installed, complete the following steps:

1. Add the following to the PATH environment variable:

   ```
   <wcs_install_path>\bin
   <db2_install_path>\sqllib\bin
   ```

2. Use the command syntax as shown in Figure 16-5 on page 454. Type the command with the applicable parameters on a command line.
3. Examine the log file to verify the results of command execution.

To use the dbclean utility on a remote database server, complete the following steps:

1. Install JDK 1.2.2 on the machine that you would like to run dbclean from.
   - Perform a custom install with the WAS V3.5 installation CD.
   - On the Choose Application Server Components window, select **IBM JDK 1.2.2**.
   - Enter the directory where you would like to have it installed. For example, the c:\ibm\java directory.
2. Copy the following files and directories from the WCS machine to the DB2 server system:

   <wcs_install_path>\lib\wcsutilities.jar

   <wcs_install_path>\lib\wcslogging.jar

   <wcs_install_path>\lib\ibmjcefw.jar

   <wcs_install_path>\lib\ibmjceprovider.jar

   <wcs_install_path>\lib\local_policy.jar

<wcs_install_path>\lib\US_export_policy.jar

<wcs_install_path>\properties*.*

<was_install_path>\lib\xml4.jar

<wcs_install_path>\bin\dbclean.bat

<wcs_install_path>\bin\setenv.bat

3. Modify the setenv.bat file. Ensure that the SET directory lines are as follows:

```
SET WAS_HOME=c:\ibm\was
SET WCS_HOME=c:\ibm\wcs
```

   – Ensure that JAVA_HOME, WCS_JCE_CLASSPATH and DB2_DRIVER have the correct path.
   – Remove or comment out the SET WCS_PATH..line.

4. From a command window, change the current directory to c:\ibm\wcs\bin or wherever the dbclean.bat file resides.
5. Use the command syntax shown in Figure 16-5 on page 454. Type the command with the applicable parameters on a command line.
6. Examine the log file to verify the results of command execution.

Note: If you would like to run dbclean remotely from machines other than the WCS or DB2 machines, you will have to install the DB2 Administrative Client and catalog it to the WCS instance database.

16.4 Job Scheduler utility

The Job Scheduler schedules and launches jobs based on a timing scheme. A job can be defined as a WCS command scheduled to run at a specified time or interval. Multiple jobs can be scheduled to run simultaneously. The command name, frequency, server nodes to be run on, etc. are stored in database tables. In a runtime environment with multiple WCS servers, connected to one WCS database server, jobs can be scheduled to run on specific WCS server nodes.

16.4.1 Job Scheduler commands

WCS provides commands that can be used to add, clean and remove jobs from the scheduler tables.

The scheduler commands are listed as follows:

- The **AddJob** command

  The **AddJob** command sets a job for the background server to run by adding an entry to the SCHCONFIG and SCHSTATUS tables. The scheduler will run the command on behalf of the user specified by the name parameter. It will run the number of times indicated by the interval parameter, and will be retried according to the values of the attempts and delay parameters.

  ```
  http://hostname/webapp/wcs/stores/servlet/AddJob?start=2000:07:15:14:15:20&
  pathInfo='/InterestItemDisplay'&URL=basemall.jsp&name=login999&queryString=
  listId%3D3
  ```

  The above example would add entries into the scheduler tables that would run the **InterestItemDisplay** command at “2000:07:15:14:15:20”, on behalf of user login999, with a name value pair listId = 3 and on completion redirect to the basemal.jsp page.

  The syntax for the **AddJob** command is shown in Figure 16-6.

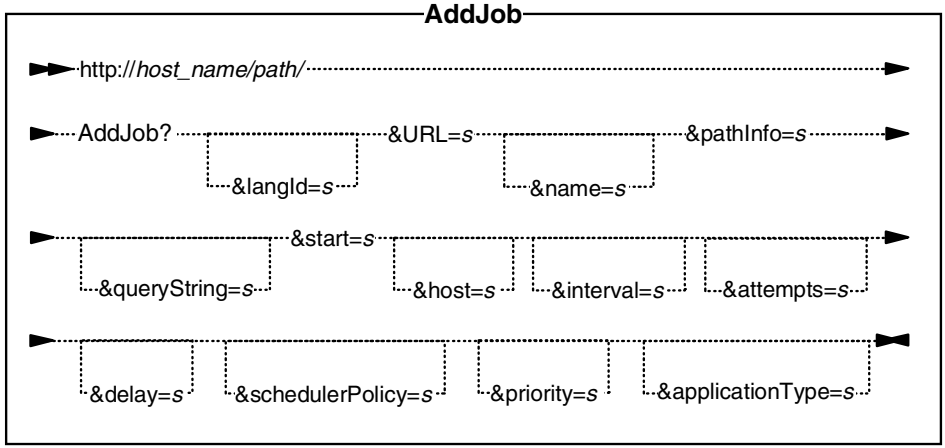


Figure 16-6 AddJob command syntax

- **CleanJob** command

  The SCHSTATUS table contains at least one record for every job in the SCHCONFIG table, and the status table is also updated with a new entry every time a job is initiated. Where jobs are executed from multiple WCS servers, or even in a single WCS server with heavy scheduler usage, the scheduler status table grows considerably large. In order to keep the scheduler running efficiently it is important to clean up the schstatus table.

  ```
  http://hostname/webapp/wcs/stores/servlet/CleanJob?endTime=2001:10:05:15:29
  :06&URL=basemall.jsp
  ```

The above command would delete all entries in the schstatus table, where the SCSEND column is less than “2001:10:05:15:29:06”.

The syntax for the **CleanJob** command is shown in Figure 16-7.

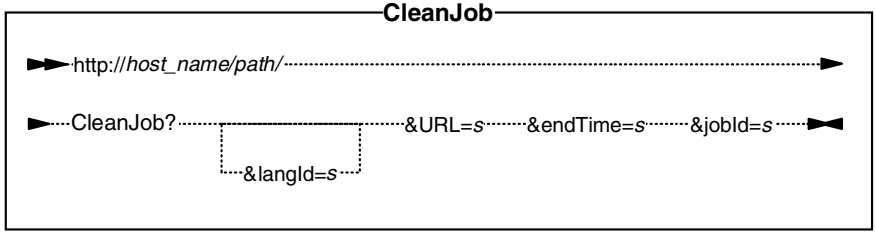


Figure 16-7 CleanJob command syntax

- The **RemoveJob** command

  The **RemoveJob** command will cancel a job that runs on the Job Scheduler. If the job is idle or in a completed/successful state, this command changes the value of the job from A to D (active to de-active). It also deletes all rows related to the job and deletes all status information related to the job. The command makes a specified job inactive. If instances of this job are currently running, the instances will be allowed to complete.

  ```
  http://hostname/webapp/wcs/stores/servlet/RemoveJob?jobId=100&URL=/demo/mal
  l.html
  ```

  Executing the above command will cancel the job having reference number 100 and if successful redirect it to mall.html.

  The syntax for the **RemoveJob** command is shown in Figure 16-6.

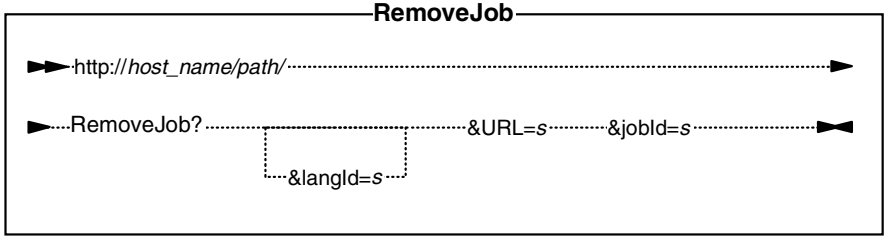


Figure 16-8 RemoveJob command syntax

- The **AddBroadcastJob** command

  The AddBroadcastJob command will be useful in an environment with multiple WCS server clones. A broadcast job applies all clones to run it immediately, or as soon as the Job Scheduler server can arrange, but only once. A broadcast job expires at a set time, which by default is 30 minutes. This can be configured in the configuration file. The configuration file resides in <wcs_install_directory>\instances\<instancename>\xml\instancename.xml. In order to change the default expiration time for a broadcast job, change the broadcastExpireTime property. Example 16-2 demonstrates the expiration time set to 45 minutes.

Example 16-2 AddBroadCastJob sample - expiration time

```
<component compClassName="com.ibm.commerce.scheduler.SchedulerComm"
               name="Scheduler"
               enable="true">
      <property cycleTime="600"
                autoClean="off"
                broadcastExpireTime="2700"
                display="false">
```

  The following command example will add entries into the SCHCONFIG and SCHSTATUS tables. This will execute the **InterestItemDisplay** command on behalf of user wcsadmin and on success redirect it to the base mall page.

```
http://hostname/webapp/wcs/stores/servlet/AddBroadcastJob?pathInfo='/Intere
stItemDisplay'&URL=basemall.jsp&name=wcsadmin
```

  The syntax for the **AddBroadcastJob** command is shown in Figure 16-9.

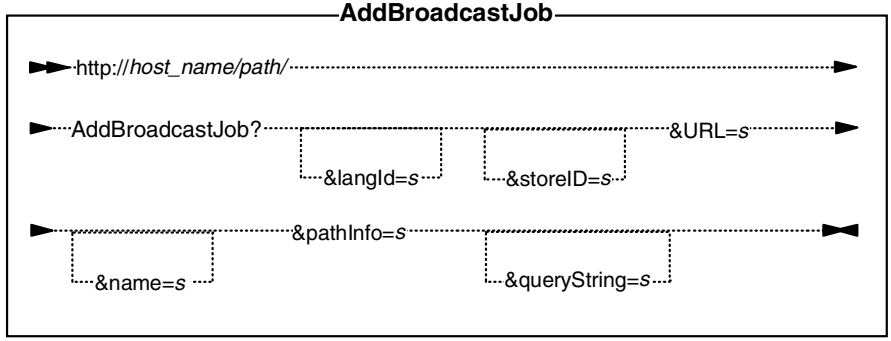


Figure 16-9 AddBroadcastJob command syntax

16.4.2 Job Scheduler tables

The tables used by the Job Scheduler are listed in the data model shown in Figure 16-10. For each job you would like to schedule, there should be a corresponding entry in both the SCHCONFIG and SCHSTATUS tables. For each entry in the SCHCONFIG table, you will have one or more entries in the SCHSTATUS table. The status of broadcast jobs is logged in the SCHBRDCST table.

The **OrderSchedule** command schedules a job that calls an internal command to create an order that is modeled after an existing order. The SCHORDERS table links the job that creates the Order (SCHORDERS.JOB_RN) and the model order (SCHORDERS.ORDER_ID). The **ScheduledOrderCancel** commands removes the entry from the SCHORDERS table.

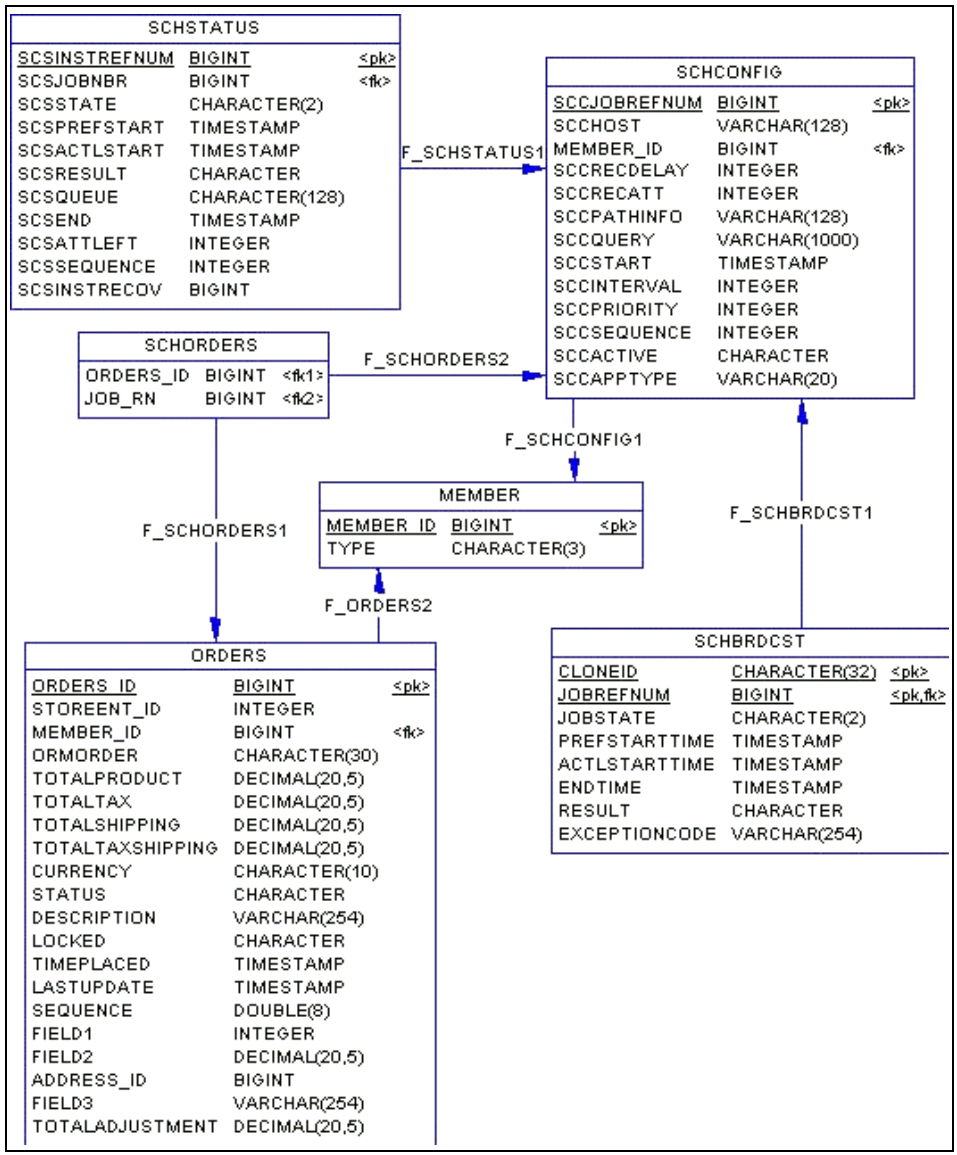


Figure 16-10 Job Scheduler data model

16.4.3 Using the Job Scheduler

Any WCS command that can be run from a browser URL can be invoked by the Job Scheduler. For a list of such commands provided by WCS, please refer to WCS online help by searching for “Commands Issued from the URL”.

Some of the commands that must be run by the scheduler and some others that are good candidates to be scheduled are discussed below.

- Auction commands

  The Auction functionality provided with WCS is heavily dependent on the scheduler. The auction commands **MonitorAuctions**, **ProcessOpenCryBids**, **ProcessDutchBids**, **ProcessAutoBids**, **DoAuctionNotify**, **CompleteOrder**, and **FinalizeAuction** are required to be run by the Job Scheduler in order to enable auctions.

- CleanJob

  Heavy usage of the scheduler loads the schstatus table. In order to keep the Job Scheduler functioning efficiently, the status table needs to be cleaned up. The **CleanJob** command is an example of a command that can be run as a scheduled job.

- **CacheDelete**

  A majority of the WCS implementations would enable caching of catalog and product pages. As you would expect, in any store there will be times when content-related category and products will change. This would require that the cached pages be deleted and the pages be recompiled and reached.

  In a multiple WCS server environment, you could create a Broadcast job for the **CacheDelete** command.

- Custom commands

  Many e-commerce sites require the creation of custom commands, for example to periodically obtain data from legacy systems and update the WCS database or create daily, weekly or monthly custom reports. These custom commands could be run as scheduled jobs.

16.5 Logging and troubleshooting

Many WCS components have their own logging capabilities. For example:

- Installation and configuration logs

  These logs are used to verify if a component was configured correctly.

- Error logs and trace files

  These logs are used to identify and resolve problems.

- Access or transaction logs

  These logs help in monitoring usage and performance.

In this section we will look at the different logs that are available, how to enable logging, and how to use logs as a problem-solving aid.

16.5.1 Enable logging and tracing

This section describes how to enable logging and tracing.

IBM HTTP Server logging

IBM HTTP Server provides two types of logs: error and access logs.

- Error log

  Any errors encountered by the Web server will be written to an error log file. The location of this file is set in the httpd.conf file ErrorLog directive. The file name and location can be modified in the configuration file.

- Access log

  Information that helps in monitoring usage and performance is logged here. LogFormat and CustomLog directives can be used to log file name and log format.

  By default the ErrorLog directive and CustomLog directive is set to logs/error.log and CustomLog logs/access.log respectively. The absence of a slash in front of the filename indicates that it is relative to ServerRoot. Assuming ServerRoot is set to the httpserver install directory, the log files will reside in the <httpserver_installpath>\logs\<filename>.log.

WAS logging

WebSphere Application Server provides the following trace and log files. The default location of the WAS trace/logs is <was_install_path>\logs. The only difference between traces and logs is that traces need to be turned on to see output in a trace file. Logs are always enabled and generated automatically, although the loglevels can be configured.

- trace.log.ibmhttp.<date>:

  This is a trace file for the Application Server plug-in. The file directory and loglevels can be set in the bootstrap.properties file, which resides in the <was_install_path>\properties folder.

  The directives to configure the plug-in trace file are as follows:

```
ose.logs.dir=C:/ibm/WAS/logs
ose.native.log.level=ERROR|WARNING
ose.plugin.log.level=ERROR|WARNING
```

- tracefile

  The tracefile is the WebSphere admin Server log file, which provides trace entries on interactions of different WAS components with the Admin Server. The admin server tracefile can be configured in the <was install path>\bin\admin.config file by adding or modifying the traceOutput and traceString attributes.

- activity.log

  The activity on the application server with details of the classes being accessed are written to the activity.log file.

- adminserver_stderr.log

  The error thrown by the code in accessing the admin server is reported here.

- oop logs

  The out-of-process logs are created both for the admin server and application server. Information on server startup, status, and change requests are reported here.

Some of the above logs are in binary. To view these binary log files use the following command:

```
showlog binaryFilename [outputFilename]
```

The showlog.bat file resides in the <wcs_install_path>\bin directory.

WCS logging

WCS logs that will be commonly used are the Servlet Engine log, application server log and trace logs, or the WCS logging service.

- servlet.og.was-oop.<date>

  This is the out-of-process log for the WCS application server. These logs are most useful in a multi-tier environment with the Web server and the application server on separate machines.

- wcs.log

  This is the application server log that shows the startup of the application server and its components, such as EJBs, Web applications, servlets and any Java errors these components report. The errors report from the standard I/O stream are reported here.

- ecmsg_<node>_<date>.log

  WCS logging service is used for logging and tracing WCS subsystems and commands. The ecmsg.logs can be configured as shown in, Figure 16-11 and Figure 16-12, using the WCS Configuration Manager. For your changes to take effect, you have to restart the WebSphere Commerce Server - <instance> (application server) from the WAS Administrator's Console.

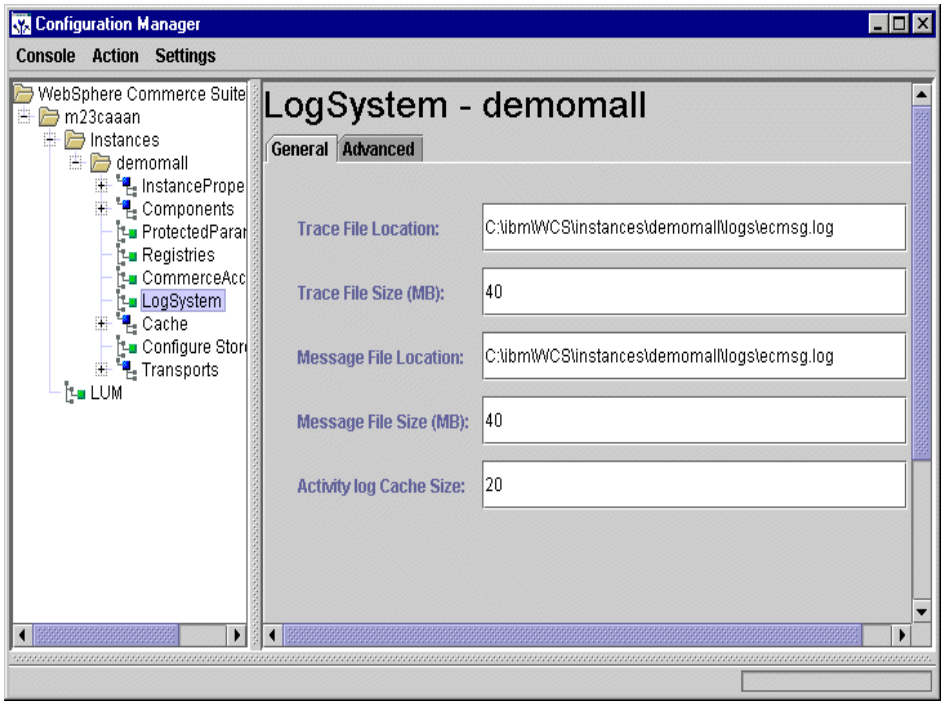


Figure 16-11 Configuration Manager - LogSystem General tab

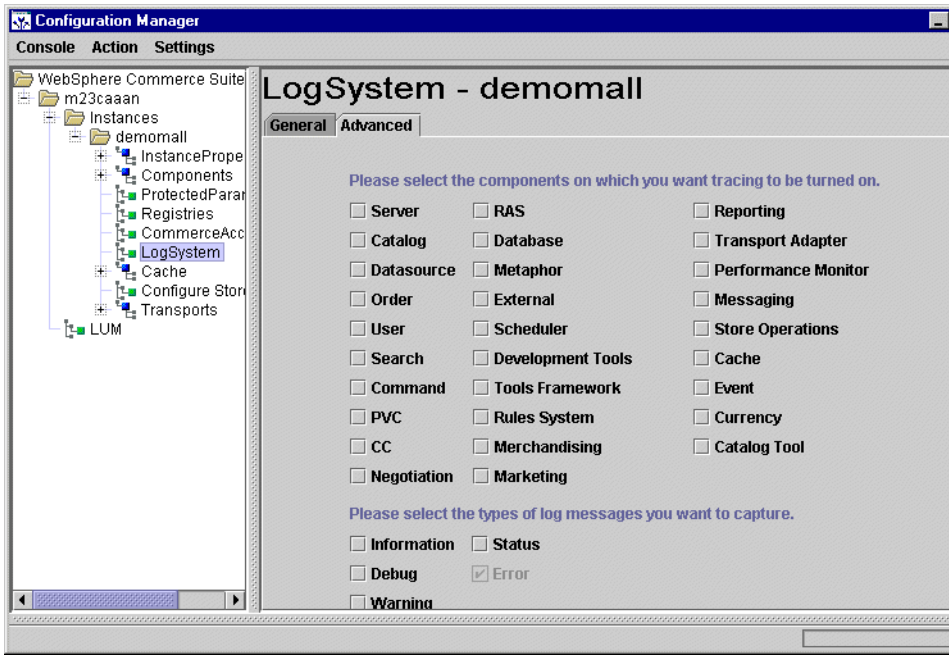


Figure 16-12 Configuration Manager - LogSystem Advanced tab

The WCS error message syntax looks as follows:

```
CMN0007S Start initializing “Tools”...
CMN0203E Could not find command”...”
```

16.5.2 Troubleshooting using log files

Problems encountered and reported in logs can be classified into four types as listed below. We will discuss the different things to look for and questions to ask in trying to solving the problem.

- Initialization failure

  The failure of a component to start is reported as an initialization failure. Things to look for in resolving the problem would be:

  - Has the component been configured correctly?
  - Are sufficient resources available to start the component?
  - Have components that the failed component is dependent on started successfully and are they responding?

- Component failure

  If a component crashes after having started, then a component failure is reported. Look for the following:

  - Are sufficient resources available?
  - What was happening when component failed?

- Connection failure

  Check if the component is configured correctly.

- Request failure

  When a request to the server fails, asking the following questions will help in diagnosing the problem:

  - Ensure proper syntax was used for the request. Check the URL and cookies if any.
  - If the URL and the name-value pairs are right, then check the database to confirm there is valid data for the parameters being passed in the URL.
  - Determine the components that should be involved in processing the request:
    - Are all the components available?
    - Are the components configured properly?
    - Does the component log provide any insight into the problem?
  - Finally check the WCS logging service to trace execution of request and identify point of failure.
  - If the failure was due to timeout, ensure sufficient resources are available.
  - If the failure was due to an authentication error, check if the ID or certificate has expired.

Part 5

Back-end integration

© Copyright IBM Corp. 2001

17

SecureWay Directory (LDAP)

This chapter describes the concepts of the Lightweight Directory Access Protocol (LDAP) and the procedure for implementing the IBM SecureWay Directory V3.21 with WebSphere Commerce Suite V5.1.

This chapter is organized into the following sections:

- Overview
- WCS V5.1 LDAP support
- SecureWay Directory V3.2 installation
- SecureWay Directory V3.2 configuration
- IBM SecureWay Directory V3.21 administration
- SecureWay Directory V3.2 SSL implementation
- Configuring WCS V5.1 for LDAP
- SecureWay Directory Management Tool (DMT)
- Working example 1
- Working example 2
- Where to find more information

© Copyright IBM Corp. 2001

17.1 Overview

In this section we provide a brief overview of the following topics:

- What is a directory?
- What is LDAP?
- Why should I use LDAP?
- What is IBM SecureWay Directory V3.2?
- Where to get SecureWay Directory V3.2

17.1.1 What is a directory?

A directory is a listing of information about objects arranged in some order that gives details about each object. Common examples of this are a city telephone directory or a library card catalog. For a telephone directory, the objects listed are people and the details given about each person are addresses and telephone numbers. Directories allow users or applications to find resources that have the characteristics needed for a particular task. For example, a directory of users can be used to look up a person's e-mail address or fax number. A directory of application servers could be searched to find a server that can access customer billing information.

17.1.2 What is LDAP?

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet these needs. LDAP defines a standard method for accessing and updating information in a directory. LDAP is gaining wide acceptance as the directory access method of the Internet and is therefore also becoming strategic within corporate intranets. It is supported by a growing number of software vendors and is incorporated into a growing number of applications. For example, the two most popular Web browsers, Netscape Navigator/Communicator and Microsoft Internet Explorer, support LDAP functionality as a base feature.

The LDAP information model is based on an entry that contains information about some object, for example a person. Entries are composed of attributes, which have a type and one or more values. Each attribute has syntax that determines what kind of values are allowed in the attribute and how those values behave during directory operations.

17.1.3 Why should I use LDAP?

People and businesses are increasingly relying on networked computer systems to support distributed applications. These distributed applications might interact with computers on the same local area network (LAN), within a corporate intranet, within extranets linking up partners and suppliers, or anywhere on the worldwide Internet. Much of this information can be shared among many applications, but it must also be protected in order to prevent unauthorized modification or the disclosure of private information.

Information describing the various users, applications, files, printers, and other resources accessible from a network is often collected into a special database that is sometimes called a directory. As the number of different networks and applications has grown, the number of specialized directories of information has also grown, resulting in islands of information that are difficult to share and manage. If all of this information could be maintained and accessed in a consistent and controlled manner, it would provide a focal point for integrating a distributed environment into a consistent and seamless system. An LDAP directory provides a centralized repository of objects that are available from any place within the enterprise.

17.1.4 What is IBM SecureWay Directory V3.2?

As organizations realize the overhead and cost involved in managing many distributed micro and macro directories, the need for a centralized solution rises. IBM understands this requirement and supports it by providing an industry-standard directory implementation using LDAP in the IBM SecureWay Directory product. The IBM SecureWay Directory product provides a general-purpose directory that multiple applications can exploit, which is very important for a successful I/T operation in medium and large environments.

IBM SecureWay Directory V3.21 uses an LDAP directory that runs as a stand-alone daemon. It is based on a client/server model that provides client access to an LDAP server. SecureWay Directory provides an easy way to maintain directory information in a central location for storage, updating, retrieval, and exchange.

IBM SecureWay Directory V3.21 provides Secure Sockets Layer (SSL) Version 3 support, both for the directory server and client. SSL provides encryption of data and authentication using X.509v3 public-key certificates. The directory may be configured to run with or without SSL support. IBM SecureWay Directory V3.21 also supports LDAP referrals, allowing directory operations to be redirected to

another LDAP directory server. Replication of the LDAP directory is supported and allows for additional copies of the directory to be available for directory read operations, which increases performance and reliability when accessing directory information.

The SecureWay Directory V3.21 alone does not provide the capability for SSL connections from LDAP clients. The SSL feature is added by installing the IBM GSKit 4.0.3.X package. The GSKit package includes Secure Socket Layer (SSL) support and associated RSA (3) technology. There are two GSKit packages available: US and Export. They come with different encryption strengths for pre-existing applications. Either package provides 56-bit encryption for new and existing applications.

Note: This redbook describes how to implement 128-bit encryption only.

SSL Support (128-bit encryption) / SSL GSKit

The IBM Global Security Kit, Version 4.0.3.X (GSKit) is an optional software package that is only required if Secure Sockets Layer (SSL) security is to be used. For SSL capability, it is necessary to install the IBM GSKit package. A version of GSKit 4.0.3.X is packaged with the SecureWay Directory V3.21 in a US domestic package and an export package.

17.1.5 Where to get SecureWay Directory V3.2

WebSphere Commerce Suite V5.1, Pro Edition includes an IBM SecureWay Directory V3.1.1.5 CD (contains DB2 V6.1). We found that the IBM SecureWay Directory V3.1.1.5 CD does not work properly with DB2 UDB V7.1, Enterprise Edition, which is required by WCS V5.1. For this reason, we recommend that you use the following with WCS V5.1:

- IBM DB2 UDB V7.1, Enterprise Edition
- IBM DB2 UDB V7.1, Fixpack 2a
- IBM SecureWay Directory V3.2

Download IBM SecureWay V3.21 from IBM

IBM SecureWay home page:

http://www.ibm.com/software/network/directory/

Download IBM SecureWay V3.21:

http://www.ibm.com/software/network/directory/downloads/

Download IBM SecureWay documentation:

http://www.ibm.com/software/network/directory/library/

17.2 WCS V5.1 LDAP support

This section is organized into the following topics:

- Overview of WCS V5.1 LDAP support
- Directories supported by WCS
- What's new in WCS V5.1 LDAP support

17.2.1 Overview of WCS V5.1 LDAP support

User profile information can be stored in either the WebSphere Commerce Suite database or an LDAP server database.

There are several authentication options in WCS:

- LDAP

  The user is authenticated against the LDAP server, using the logon ID and password provided by the user. User information on the LDAP server is replicated to the Commerce Suite database for runtime operational use.

- WCS Database

  The user is authenticated against the Commerce Suite database using the logon ID and password provided by the user.

- Other

  Authenticate using a third-party interface, and store profile data in the WebSphere Commerce Suite database.

The authentication mode can be set on the member subsystem page of the WCS Configuration Manager. X.509 certificates can be used with either LDAP or database authentication.

If X.509 certificates are used, the Web server authenticates the user. In this case, the authentication mode setting determines where profile data is stored, LDAP or the WCS database.

17.2.2 Directories supported by WCS

WCS V5.1 supports the following LDAP directories:

- IBM SecureWay Directory Server V3.2
- Netscape Directory 4.1.2

Note: Implementing Netscape Directory is not covered in this book.

For more information on WCS and Netscape Directory refer to the WCS online help, or the installation guide for the given WCS V5.1 operating system platform.

Netscape Directory information can be found at:

- http://www.iplanet.com
- http://www.iplanet.com/download_index/downloads_index_9_0.html

17.2.3 What's new in WCS V5.1 LDAP support

This section includes new LDAP design features included in WCS V5.1, as well as some limitations.

LDAP design features

The following information highlights some of the design features of the LDAP implementation in WebSphere Commerce Suite V5.1:

- Multiple WCS attributes can be replicated to a single LDAP attribute by specifying a separator character.
- The three address attributes ADDRESS1, ADDRESS2, ADDRESS3 (analogous to saadr1/saadr2/saadr3 in WCS V4.1) and the three name attributes FIRSTNAME, LASTNAME, and MIDDLENAME (analogous to safname/samname/salname in WCS V4.1) are no longer treated as special cases.
- Multiple LDAP attributes can be replicated to a single WCS attribute. To do this use the multiple <map> sections in the ldapmap.xml file; however, you should be aware that overwrites can occur.
- A single WCS attribute can be replicated to multiple LDAP attributes.
- A single LDAP attribute can be replicated to multiple Commerce Suite attributes.
- Only the default address for the user is persisted to LDAP. The default registration address is defined as the address in the user's address book with selfaddress=1 and logonid=nickname, where logonid is the logon ID in the USEREG table and nickname is NICKNAME in the ADDRESS table. Other addresses in the user's address book will not be stored on LDAP.
- WCS V5.1 allows administrators to be persisted to LDAP (WCS V4.1 did not allow this). This can be accomplished as follows:
  - Extend the LDAP schema to create an auxiliary class.

- The auxiliary class can contain a single attribute that specifies a role, such as roleName.
- Specify the auxiliary class as one of the classes to be used for LDAP persistence in the Configuration Manager (see LdapPersonOCS).
- Modify the ldapmap.xml file to map the LDAP attribute in the new auxiliary class to the WCS attribute RegisterType in the User access bean.

Note: The RegisterType in the user access bean can have one of two valid values with respect to roles: 'S' for Site Administrator and 'A' for Administrator.

- Only registered users from WCS will be persisted to LDAP, not guest users.

Limitations

Table 17-1 describes the limitations of LDAP support in WCS V5.1.

Table 17-1 WCS V5.1 LDAP support limitations

Limitation	Description
Multi-valued attributes in LDAP	WCS does not use multi-valued attributes; however, multi-valued attributes in LDAP are tolerated. In Commerce Suite Version 5.1, if the new value already exists on LDAP, no update is performed; otherwise, the first value of the multi-valued LDAP attribute is updated.
LDAP referrals and references	WCS does not support LDAP referrals and references.

Limitation	Description
Multiple user entries found during search	WCS allows multiple search roots to be specified when searching for a user entry on LDAP. The search roots are specified using the LdapPersonSearchRoot attribute. The LdapPersonDefaultBase value should always be under one of the search roots and will always be searched first. If multiple user entries match the search criteria, the "first" entry returned by the LDAP server will be used. The search criteria is formed as follows: LdapPersonRDN=value,searchRoot where: ► LdapPersonRDN is the LDAP attribute name used as the RDN attribute, for example, uid. ► value is – The user's logon ID If logon ID and password are configured to be the authentication challenge type – The X.509 certificate subject name If X.509 is configured to be the challenge type ► searchRoot Is one of the values of LdapPersonSearchRoot. For performance reasons, the search roots specified in the LdapPersonSearchRoot attribute should not overlap; otherwise, some users may be searched for more than once. It is not necessary to specify search roots where one is covered by another.
Multi-component RDN	Only one LDAP attribute can be specified as the RDN attribute, not a combination of LDAP attributes.

17.2.4 LDAP authentication scenarios

In addition to using LDAP with WCS V5.1, you may already be using or plan to use an LDAP server with other applications. To avoid situations where an existing LDAP user is unable to log on to WCS, or where a WCS user is no longer able to log on to WCS, you must ensure that Logon IDs are unique for all users under all search roots. This section describes how users are created and authenticated using LDAP.

Example scenario: create a user on an LDAP server

A user can be created on an LDAP server by completing the following high-level steps:

- Registering through WCS
- Using the LDAP interface
- Using applications other than WCS connected to the same LDAP server

If the user entry is created by registering through WCS, WCS does the following:

- Ensures that the LogonId is unique within WCS
- Ensures that the LogonId is unique under all search roots as specified by the LdapPersonSearchRoot parameter
- Creates the user under the LDAP entry with DN specified by LdapPersonDefaultBase parameter
- Creates the user entry in the WCS database

Regardless of how the user entry was created on LDAP, the LDAP user can log on through WCS, provided that the LdapPersonSearchRoot is configured with the proper values. If authentication is successful, the user becomes, if he was not already, a WCS user after logon. The user's information is replicated to the WCS database.

When a WCS user who does not yet exist on LDAP (there is no user entry under all the search roots with the same LogonId as the user in the WCS database) logs on to WCS, WCS does the following:

- Attempts to authenticate the user by searching the LDAP server for the user entry. Since the user entry does not yet exist, authentication will fail.
- Attempts to authenticate the user using the WCS database and if successful, the user entry is migrated “on-the-fly” from WCS to LDAP.

For more information about two common authentication scenarios, refer to our working examples found in 17.9, “Working example 1” on page 525 and 17.10, “Working example 2” on page 528.

17.3 SecureWay Directory V3.2 installation

This section describes how to install IBM SecureWay Directory V3.2 for AIX and prerequisite components.

The IBM SecureWay Directory V3.2 for AIX installation is organized into the following phases:

- SecureWay Directory V3.2 runtime environment
- AIX 4.3.3 and prerequisite file sets installation
- IBM JDK 1.1.8 PTF 10 installation
- Netscape Navigator installation
- IBM HTTP Server V1.3.12 installation
- IBM DB2 V7.1 Server installation
- IBM SecureWay Directory V3.2 installation

17.3.1 SecureWay Directory V3.2 runtime environment

This section provides information on the SecureWay Directory V3.2 runtime environment security considerations, hardware, software, and high-level steps for installation.

Security considerations

The LDAP server that is provided with the domestic version of the SecureWay Directory Version 3.21 is enabled for 128-bit and triple-DES encryption, as are the client utilities (ldapsearch, ldapmodify, etc.). The LDAP server that is provided with the export version of the SecureWay Directory Version 3.21 is enabled for 56-bit DES encryption, as are the client utilities (ldapsearch, ldapmodify, and so forth).

Note: For more information on security, refer to the client.txt file CD and the *C Programming Reference* (found on the product CD) for topics such as:

- Developing LDAP applications to use the stronger encryption algorithms.
- Upgrading existing LDAP applications so that they continue to have access to strong encryption.

In order to install a secure LDAP directory using SecureWay Directory V3.2, first install SecureWay Directory V3.2. After the installation is complete, install the GSKit 4.0.3.X found in the SecureWay Directory V3.21 package (CD or download).

Note: SecureWay Directory V3.2 does works without the GSKit installed. However, it will only accept non-SSL connections from LDAP clients.

17.3.2 Hardware and software prerequisites

For detailed information on hardware and software prerequisites, refer to the *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*.

17.3.3 Hardware used in our test environment

We used the following hardware in our test environment for the IBM SecureWay Directory V3.2 for AIX server:

- 1 x IBM RS/6000 43P Model 150 with 768 MB of memory and 8 GB hard disk (hostname: jupiter.itso.ibm.com).

The WebSphere Commerce Suite V5.1 3-tier runtime environment is installed on separate systems. The hostname of the Commerce/WebSphere Application Server system is m23bzzpg.itso.ral.ibm.com. We will refer to this system as the WCS server.

17.3.4 Software used in our test environment

We used the following software in out test environment for the IBM SecureWay Directory V3.2 for AIX server:

- IBM AIX V4.3.3 + (prerequisite file sets + maintenance level 6)
- IBM HTTP Server V1.3.12
- IBM DB2 UDB V7.1 + Fixpack 2a
- IBM JDK V1.1.8 + PTF 8

  Note: The SecureWay Directory V3.21 does not require an external Java installation. The product installs a private version (JRE 1.2) for its own use.

- IBM SecureWay Directory V3.2 for AIX

We used the following ports for LDAP:

  - 389 (standard)
  - 636 (secure)

High-level installation steps

When installing IBM SecureWay Directory V3.2 for AIX, we recommend an incremental installation approach (for example, install and test each component individually). Table 17-2 provides the high-level steps for installing IBM SecureWay Directory V3.2 for AIX.

Table 17-2 IBM SecureWay Directory V3.2 for AIX: high-level installation steps

Target system	Installation step	Description	Result / verification
AIX - LDAP server	Install AIX, prerequisite file sets, and service	Mandatory step	Check installed file sets via smitty
AIX - LDAP server	Install IBM JDK 1.1.8 PTF 10	Mandatory step	Check installed file sets via smitty and in a terminal window type: `# java -fullversion`
AIX - LDAP server	Install Netscape Navigator 4.07 or higher	Optional - but recommended for easy installation verification	Try to request `http://www.ibm.com`
AIX - LDAP server	Install the IBM HTTP Server	Mandatory step	Check installed file sets via smitty and try to request `http://<your_hostname>/back.gif`
AIX - LDAP server - optional -	Configure the IBM HTTP Server for SSL	Optional - but highly recommended	Check installed file sets via smitty and try to request `https://<your_hostname>/back.gif`
AIX - LDAP server	Install DB2 V7.1 Server	Mandatory step	Check installed file sets via smitty
AIX - LDAP server	Install the DB2 V7.1 Fixpack 2a	Mandatory step	Check installed file sets via smitty
AIX - LDAP server	Install SecureWay Directory V3.2	Mandatory step	Check installed file sets via smitty and try to request `http://<your_fully_qualified_hostname>/ldap`
AIX - LDAP server - optional -	Install SecureWay Directory V3.21 SSL support	Optional - but highly recommended	Check installed file sets via smitty
AIX - LDAP server	Installation verification	Optional - but highly recommended	Test 128-bit with Navigator Address Book
WCS server	Configure WCS for SecureWay V3.2	Mandatory step	Perform the working examples

Target system	Installation step	Description	Result / verification
WCS server - optional -	Configure WCS for SecureWay V3.21 SSL support	Optional - but highly recommended	Perform the working examples
AIX - LDAP server & WCS server	Overall system test	Optional - but highly recommended	Perform the working examples

Note: For guidance on implementing IBM SecureWay Directory V3.2 for AIX in different environment or using different components, refer to the product documentation.

17.3.5 AIX 4.3.3 and prerequisite file sets installation

After installing AIX 4.3.3 on the SecureWay Directory V3.2 server system, prerequisite AIX file sets and service levels must be applied.

Prerequisite AIX file sets

After the installation of AIX 4.3.3, ensure that the AIX file sets listed in Table 17-3 are installed.

Table 17-3 AIX prereqs for WAS V3.5

File set	Description of file set	AIX 4.3.3 CDs
bos.adt.base	Base Application Development Toolkit	CD-ROM Vol 1
bos.adt.include	Base Application Development Include Files	CD-ROM Vol 1 and 3
X11.adt.lib	AIXwindows Application Developer's Toolkit Libraries	CD-ROM Vol 3
X11.adt.motif	AIXwindows Application Developer's Toolkit Motif	CD-ROM Vol 3

Install AIX 4.3.3 service level 6

After you have installed the required AIX 4.3.3 file sets for SecureWay Directory V3.2, install AIX service level 6. For details refer to"Install AIX V4.3.3 maintenance level 6" on page 144.

Verify prerequisite file levels

After the AIX 4.3.3 service level 6 installation and restart of your system, ensure that the level of the file sets is equal to or greater than the values shown in Table 17-4.

Table 17-4 AIX prerequisite file set levels for SecureWay Directory V3.2

APAR Number	Level	PTF Number	File set
IY03993	4.3.3.1	U467183	bos.adt.include
IY06365	4.3.3.1	U467290	bos.net.tcp.client
IY04069	4.3.3.3	U467478	bos.sysmgmt.serv_aid
IY05690	4.3.3.2	U467572	X11.base.lib
IY05697	4.3.3.2	U467473	X11.adt.motif
IY05741	4.3.3.1	U467558	X11.base.rte
IY05989	4.3.3.2	U467459	X11.Dt.rte
IY05989	4.3.3.3	U467557	X11.motif.mwm
IY05990	4.3.3.2	U467458	X11.motif.lib
IY05990	4.3.3.2	U467616	X11.compat.lib.X11R5
IY06171	4.3.3.3	U467283	bos.rte.libpthreads
IY06171	4.3.3.3	U467444	bos.adt.prof
IY06121	4.3.3.2	U467222	X11.Dt.lib

Additionally, you need one of the APARs listed in Table 17-5 installed, depending on whether your system is a uniprocessor or a multiprocessor.

Table 17-5 AIX prerequisite file set levels: uni/multi-processor

	APAR Number	Level	PTF Number	File set
Uniprocessor	IY06625	4.3.3.3	U467275	bos.up
Multiprocessor	IY06625	4.3.3.3	U467531	bos.mp

These APARs can be obtained from the IBM RS/6000 support Web site at:

http://service.software.ibm.com/cgi-bin/support/rs6000.support/downloads

Alternatively, you can download the fixes from http://service.software.ibm.com/rs6k/fixdb.html or use the AIX 4.3.3 Update CD (October 2000).

Note: After you apply all the services that you need for your system, restart your system to enable the changes.

17.3.6 IBM JDK 1.1.8 PTF 10 installation

This section describes how to install the IBM JDK 1.1.8 PTF 10. AIX 4.3.3 installs the JRE 1.1.8 by default. If you do not have JRE 1.1.8 installed, install it from the AIX 4.3.3 CDs.

Note: You can identify the installed version of IBM JDK by typing:

```
# java -fullversion
```

Create a download directory for JDK 1.1.8 PTF 10

To create a download directory for the JDK 1.1.8 PTF 10, complete the following steps:

1. Create a directory

   ```
   # mkdir -p /sw/jdk118.ptf10
   ```

2. It may be necessary to increase the free disk space on your hard drive. Before increasing the file system size, view the existing free space by using the **df** command:

   ```
   # df
   ```

3. Calculate the size required:

   new_size = current_size + (desired_free_space - free_space)

4. Increase the size:

   ```
   # chfs -a size='<new_size>' /<file_system_mount_point>
   ```

5. Verify file system free space:

   ```
   # df
   ```

Download JDK 1.1.8 PTF 10

This section describes how to download the JDK 1.1.8 PTF 10.

1. Make sure you download the following file sets:
   - Java.adt.includes.1.1.8.6
   - Java.adt.src.1.1.8.10

- Java.rmi-iiop.docs.1.1.8.10
- Java.rmi-iiop.lib.1.1.8.10
- Java.rte.bin.1.1.8.10
- Java.rte.classes.1.1.8.10
- Java.rte.lib.1.1.8.10
- Java.security.lib.1.1.8.10

2. From a Web browser enter the following URL to obtain JDK 1.1.8 PTF 10:

   `http://service.software.ibm.com/rs6k/fixdb.html`

3. An AIX Fix Distribution window appears, Enter JDK 1.1.8 in the search field and click **Find Fix**.
4. When the Results from search window appears, select **JDK 1.1.8 PTF 10** and click **Get Fix Package** (see Figure 17-1).

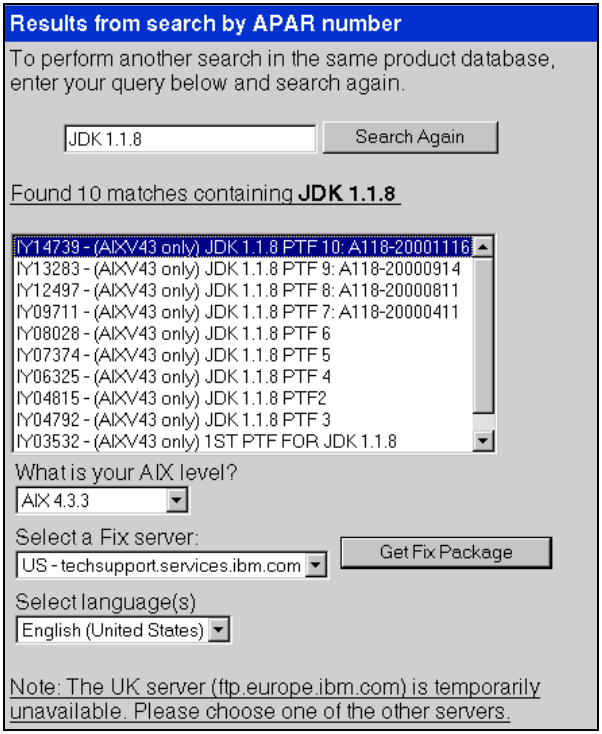


Figure 17-1 JDK 1.1.8 PTF 10 download

5. The Fix Package from IBM Corporation page appears.

Note: You will also find there the total byte size of the fix package.

Install JDK 1.1.8 PTF 10

To get detailed installation instructions, click **Installation Instructions - Please read** on the Fix Package from IBM Corporation page.

The AIX Version 4 Update Installation Instructions page appears and guides you through the installation process.

To install JDK 1.1.8 PTF Level 10, complete the following steps:

1. Start the install as follows:

```
# cd /sw/jdk118.ptf10
# smitty install_all
```

2. When the Install and Update from All Available Software window appears, enter `./` in the Input device/directory for software field and then press Enter.
3. In the Software to Install field, press F4 for a list and highlight the following packages using F7 to select, and then press Enter:
   - Java.adt
   - Java.rmi-iiop
   - Java.rte
   - Java.security
4. When the Install and Update from All Available Software window appears, press Enter.
5. When the `Are You Sure` message is displayed, press Enter to install.
6. Scan the log file to verify that the file sets were installed successfully.
7. Press F10 to return to the system prompt.

Verify the JDK 1.1.8 + PTF Level 10

To verify that the PTF 10 is installed, type the following:

```
#java -fullversion
```

The following message should be displayed:

```
java full version "JDK 1.1.8 IBM build a118-20001116 (JIT enabled: jitc
V3.1-JDK1.1-20001116
```

17.3.7 Netscape Navigator installation

SecureWay Directory V3.2 requires Netscape Navigator V4.07 or higher for administration. Netscape Navigator can be obtained from one of the following:

- AIX Bonus Pack CD
- http://www.netscape.com

Refer to the product documentation for installation instructions.

Note: In our working example, we used Netscape Navigator V4.73i for AIX, and V4.75 for Windows NT. Both were English versions with 128-bit encryption strength. Make sure you have these or equivalent versions available, because there may be some issues about other versions.

17.3.8 IBM HTTP Server V1.3.12 installation

For instructions on installing the IBM HTTP Server V1.3.12 for AIX, refer to 7.2, “Install the IBM HTTP Server” on page 144.

Note: Please make sure that you *do not select* the following LDAP modules during installation:

- http_server.modules.ldap
- http_server.modules.ldap128

We recommend that you configure SSL support for the IBM HTTP Server, as documented in 7.2.3, “Configure the HTTP Server” on page 147.

17.3.9 IBM DB2 V7.1 Server installation

For instructions on installing the IBM DB2 UDB V7.1, EE for AIX, refer to 7.3, “Install the DB2 Server” on page 151.

Included in the instructions is the procedure to install the DB2 V7.1 Fixpack 2a.

Note: Disregard the section on creating a WAS database.

Important: For AIX users with DB2 7.1 the libdb2.a library needs to be re-linked as follows after a SecureWay installation:

```
# cd /usr/ldap/lib
# rm libdb2.a
# ln -s /usr/lpp/db2_07_01/lib/libdb2.a libdb2.a
```

17.3.10 IBM SecureWay Directory V3.2 installation

This section describes the installation procedure for IBM SecureWay Directory V3.2 for AIX.

17.3.11 Install SecureWay Directory V3.2

To install IBM SecureWay Directory V3.2 for AIX, complete the following steps:

1. Ensure that you have installed the installation prerequisites (especially that the GSKit is installed correctly).
2. Log in to AIX as user ID root, and start an AIX terminal session.
3. Insert the SecureWay Directory CD into the CD-ROM drive or change to the download directory.
4. Mount the CD-ROM:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   # cd /mnt/usr/sys/inst.images
   # smitty install_all
   ```

5. When the Install and Update from All Available Software window appears, enter `./` in the Input device/directory field and then press Enter.
6. In the Software to Install field, press F4 for a list and highlight the packages in Table 17-6, using F7 to select.

Table 17-6 SecureWay Directory V3.2 installation file sets

File set	Description
ldap.client	SecureWay Directory client
ldap.html.<locale>	HTML install, config guides, and manual; we selected ldap.html.en_US for U.S. English.
ldap.msg.<locale>	SecureWay Directory messages. We selected ldap.html.en_US for U.S. English.

File set	Description
ldap.server	SecureWay Directory server and administration interface.
ldap.max.crypto_client	SecureWay Directory client - upgrade for 128-bit encryption strength.
ldap.max.crypto_server	SecureWay Directory server and administration interface - upgrade for 128-bit encryption strength.

7. When you finish selecting the file sets, press Enter.
8. When the `Are You Sure?` message appears, press Enter.
9. Check the installation summary at the end of the output to verify the successful installation of the file sets.
10. Press F10 to return to a system prompt.

Verify the SecureWay Directory V3.2 installation

Make sure the following file sets exist on your system showing the appropriate level 3.2.0.0 by typing:

```
# lslpp -l |grep ldap
```

You should see the following:

```
ldap.client.adt
ldap.client.rte
ldap.html.en_US.config
ldap.html.en_US.man
ldap.max.crypto_client.adt
ldap.max.crypto_client.rte
ldap.max.crypto_server.admin
ldap.max.crypto_client.com
ldap.msg.en_US
ldap.server.admin
ldap.server.com
ldap.server.rte
```

For AIX users with DB2 7.1, the libdb2.a library needs to be re-linked as follows:

```
# cd /usr/ldap/lib
# rm libdb2.a
# ln -s /usr/lpp/db2_07_01/lib/libdb2.a libdb2.a
```

You have successfully installed IBM SecureWay Directory V3.2 for AIX.

17.4 SecureWay Directory V3.2 configuration

The SecureWay Directory server can be configured by using a graphical user interface (GUI) or by a command line utility.

17.4.1 Configure SecureWay Directory using a GUI

This section provides instructions for configuring SecureWay Directory using the configuration utility. The SecureWay Directory V3.2 configuration utility can be used to configure the following:

- Directory administrator distinguished name (DN) and password
- Configure a database
- Configure a Web server

You can select one of these tasks or select multiple tasks. If you select more than one task, the information entry windows are displayed consecutively.

Note: If you already have a SecureWay Directory database, make sure you select **Use existing database** when prompted.

In this working example, we selected all three options. To configure a new SecureWay Directory system, complete the following steps:

1. Log in to AIX as user ID root, and start an AIX terminal session.
2. Start the configuration utility by typing the following:

   ```
   # ldapxcfg
   ```

3. When the SecureWay Directory Configuration welcome window appears, select the checkbox for each of the options as seen in Figure 17-2 and then click **Next**.

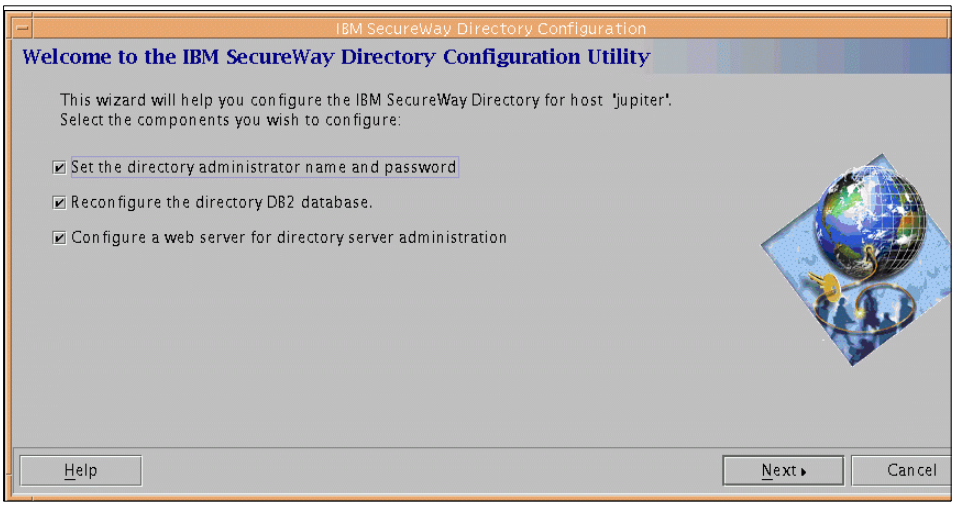


Figure 17-2 SecureWay Directory V3.2: configuration utility options

4. Enter the directory administrator distinguished name (DN) and password, as seen in Figure 17-3, then click **Next**.

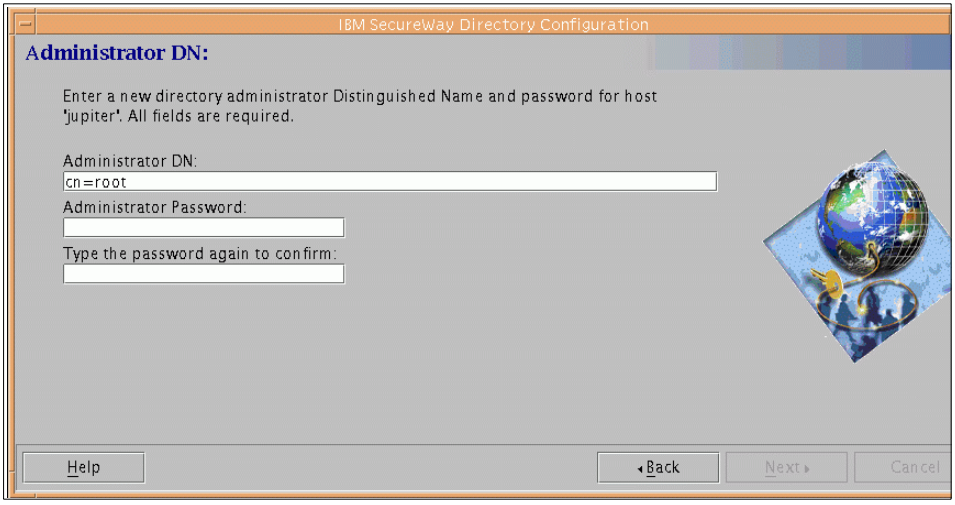


Figure 17-3 SecureWay Directory V3.2: configuration utility admin DN

5. The Configure DB2 database window appears as seen in Figure 17-4. Select **Create a default LDAPDB2 database,** then click **Next**.

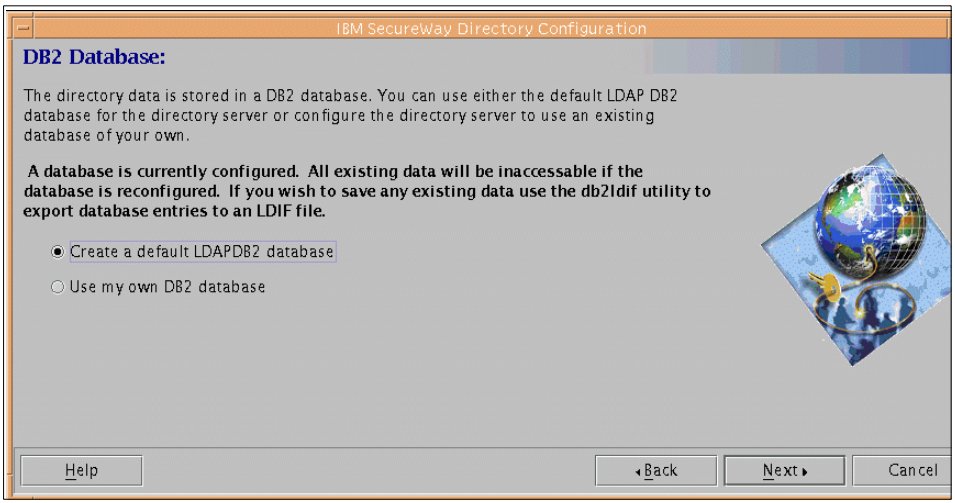


Figure 17-4 SecureWay Directory V3.2: configuration utility create database

6. When the Create the database using local codepage or UTF-8 window appears, select **Create a Universal DB2 database (UTF-8)** as seen in Figure 17-5, then click **Next**.

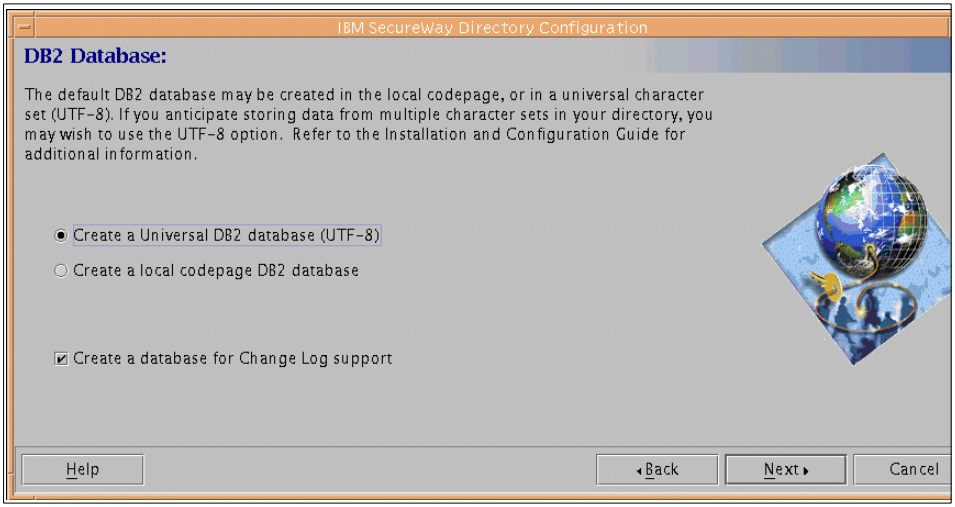


Figure 17-5 SecureWay Directory V3.2: configuration utility create UTF-8 database

7. Enter the database directory path to store the LDAPDB2 database as seen in Figure 17-6, then click **Next**.

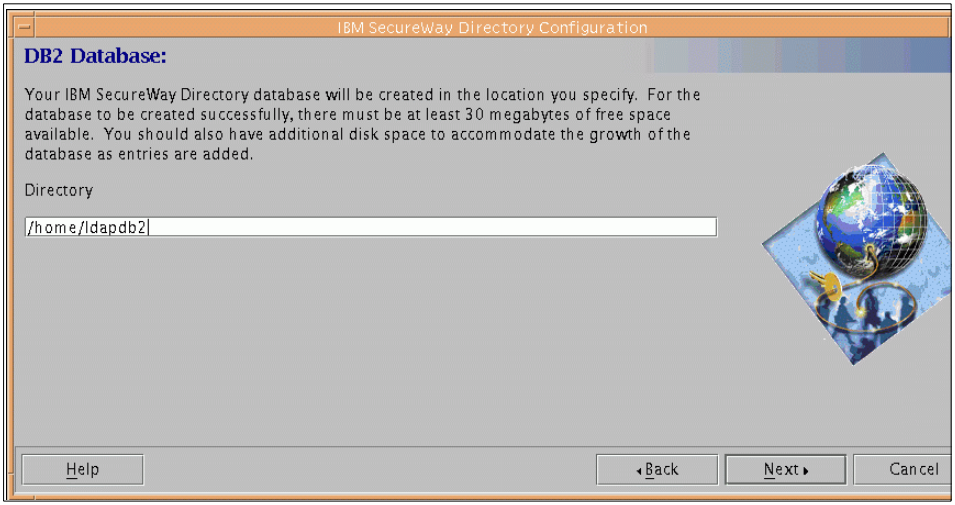


Figure 17-6 SecureWay Directory V3.2: configuration utility database path

8. When the Web server to configure window appears, select **IBM HTTP** as seen in Figure 17-7, then click **Next**.

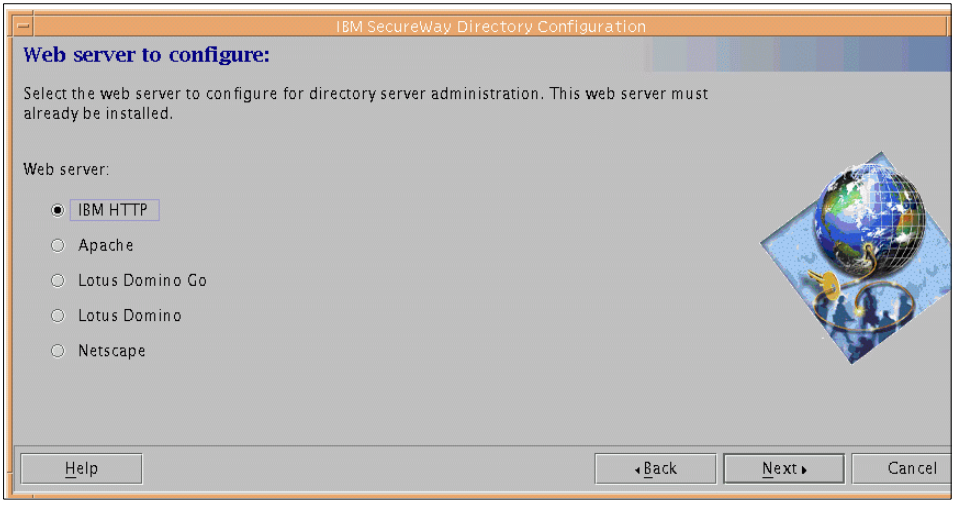


Figure 17-7 SecureWay Directory V3.2: configuration utility Web server selection

9. Enter the full path name of the Web server configuration file as seen in Figure 17-8. For the IBM HTTP Server V1.3.12.0 enter the following:

```
/usr/HTTPServer/conf/httpd.conf
```

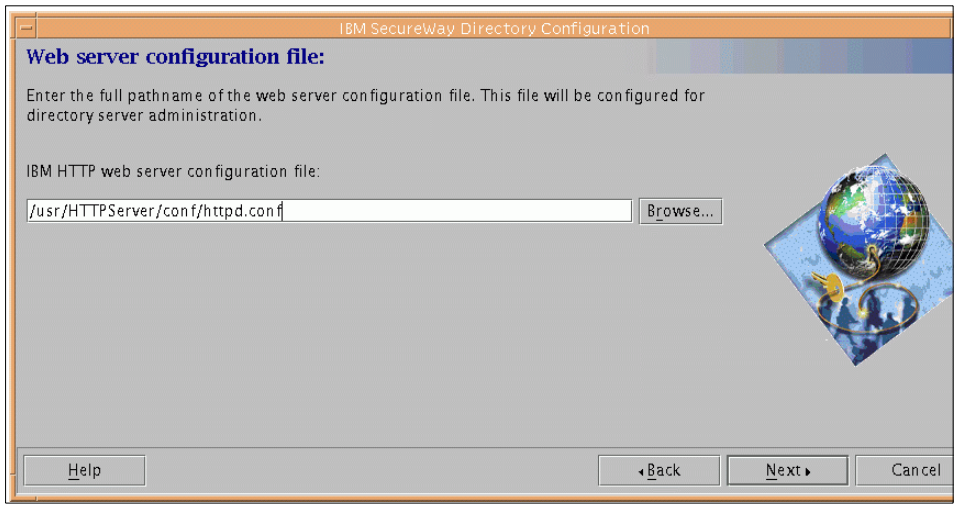


Figure 17-8 SecureWay Directory V3.2: configuration utility Web server path

10. When the Review the Configuration Summary window appears, as seen in Figure 17-9, click **Configure**.

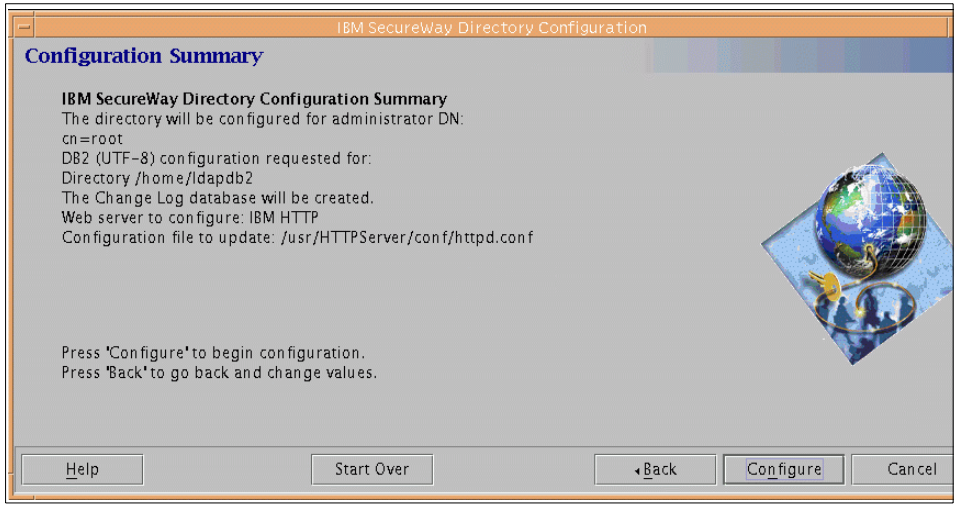


Figure 17-9 SecureWay Directory V3.2: configuration utility summary

11. When the Configuration Completion window appears, make a note of the administration URL, then click **OK**.
12. Restart the configured Web server.

    For the IBM HTTP Server V1.3.12.0 type the following:

```
# cd /usr/HTTPServer/bin
# ./apachectl restart
```

13. To start the SecureWay Directory V3.2 Server from the command line, type the following:

```
# cd /usr/ldap/bin
# ./slapd
```

Note: The spelling of the command above is correct (slapd).

17.5 IBM SecureWay Directory V3.21 administration

The SecureWay Directory V3.2 Server Administration GUI is a Web-based administration utility that supports among other things the following administrative tasks:

- Start and stop the server
- Add and delete suffixes
- Add entries to the directory database
- Configure a database
- Configure a replica
- Define an ACL

17.5.1 Start and stop the server

To start the SecureWay Directory Server Administration GUI, complete the following steps:

1. Start the SecureWay Directory Server Administration GUI by entering the following URL in a Web browser:

   `http://<hostname>/ldap`

2. When the BM SecureWay Directory Server Administration Logon window appears, as seen in Figure 17-10, enter the following and then click **Logon**.
   - Admin ID: `cn=root`
   - Password: `<your_password>`

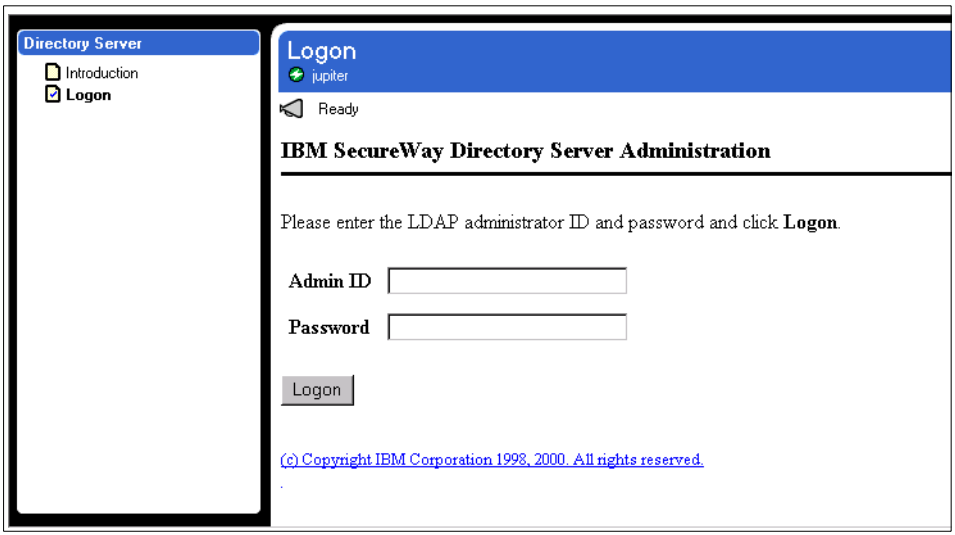


Figure 17-10 SecureWay Directory Server Administration: logon

3. From the left-hand frame, select **Server -> Startup/Shutdown -> Startup**. When the startup is finished, a completion window is displayed.

17.5.2 Add and delete suffixes

Before entries can be added to an LDAP directory, a suffix must be defined for that directory. A suffix specifies the distinguished name (DN) for the root of a directory, and is the highest entry stored in the directory for a server.

Each entry stored ends with this suffix, and each entry to be added to the directory must have a suffix that matches one of the defined suffixes on the server. An LDAP server can house multiple suffixes, which can be added and deleted from the directory.

Add a suffix

To add a suffix to a directory, complete the following steps:

1. Start the SecureWay Directory Server Administration GUI by entering the following URL in a Web browser:

   `http://<hostname>/ldap`

2. When the IBM SecureWay Directory Server Administration Logon window appears, as seen in Figure 17-10, enter the following and then click **Logon**.
   - Admin ID: `cn=root`
   - Password: `<your_password>`

3. From the left navigation frame, click **Suffixes**.
4. Click **Add a suffix**.
5. To add a suffix for this server, enter the following and then click **Add a new suffix**:
   - Suffix DN: `o=ibm,c=us` (working example)

7. Restart the SecureWay Directory Server:

   From the left-hand frame of the Directory Server Administration GUI, click the upper right icon next to the question sign.

Delete a suffix

To delete a suffix from a directory, complete the following steps:

1. Click **Delete suffixes**
2. Click the box next to the suffix you want to delete.
3. Click the **Delete suffixes** button.
4. Restart the SecureWay Directory Server:

   From the left-hand frame of the Directory Server Administration GUI, click the upper right icon next to the question sign.

5. You can verify your operation by clicking **Suffixes** -> **List suffixes** as seen in Figure 17-11.

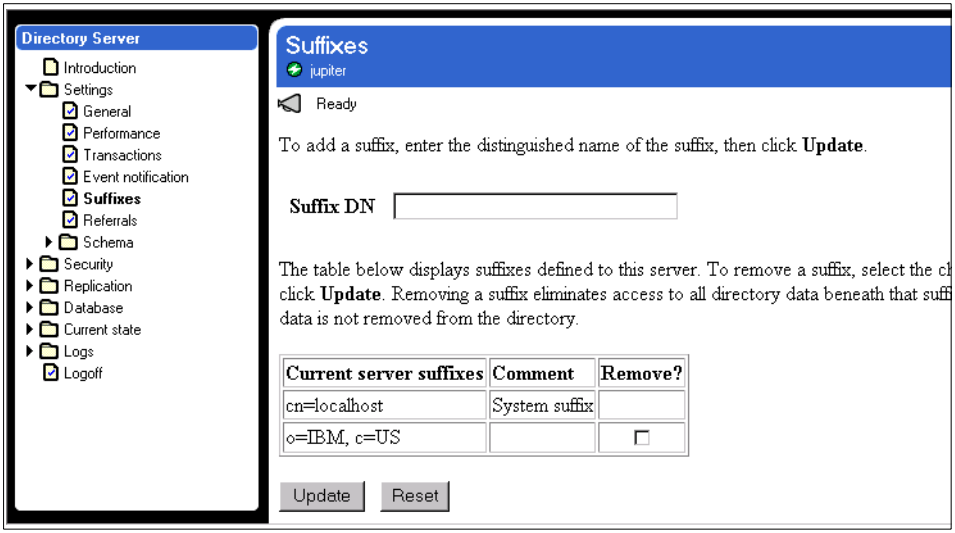


Figure 17-11 List of suffixes

Note: Removing a suffix will disable access to all directory data underneath that suffix. This does not physically remove the data from the directory. Access can be restored to a deleted suffix by adding it again. The suffixes are recorded in the slapd.conf file (we do not recommend that you manually edit this file).

17.5.3 Add entries to the directory database

There are two methods of adding entries to the directory database:

1. LDIF (LDAP Data Interchange Format) data file

   LDIF is a standard format for representing LDAP entries in text form and is used to import entries into the database.

2. SecureWay Directory Management Tool (DMT)

   The SecureWay Directory Management Tool provides a GUI to add entries to the database. For more information, refer to 17.8, “SecureWay Directory Management Tool (DMT)” on page 516.

Note: Prior to adding entries to the database a suffix (DN) must exist on the server, and all entries to be added must have a suffix that matches the DN value.

Adding data entries to the database using LDIF

There are two ways to import an LDIF file, either through the SecureWay Directory Server Administration GUI or by using command line utility. For the working example we have provided step-by-step instructions for using the SecureWay Directory Server Administration GUI to import the LDIF file.

LDIF import - SecureWay Directory Server Administration GUI

For the example below, the sample LDIF file shipped with the product was used. Some editing is usually necessary to adapt the LDIF file to your environment (for example, to define the correct suffix).

To add data entries to the database using LDIF with a GUI, complete the following steps:

1. Start the SecureWay Directory Server Administration GUI, and log on.
2. From the left navigation frame, click **Database** -> **Add entries**.
3. To add entries to the database from an LDIF file, specify an LDIF file. In this example, we use the sample.ldif file located in /usr/ldap/examples.

We decided to use the sample.ldif file for demonstration purposes. The contents of this file are 46 user entries.

Note: The LDAP user ID must have proper AIX permissions in the LDIF file prior to the import into the database in the next step.

4. Enter the location of the file containing the entries in the text entry field and click **Add entries to database**.

   This loads the LDIF file into the directory. If successful, you should see the message below:

   ```
   The directory was successfully updated.
   ```

   Remember that o=IBM,c=US suffix must be added *before* importing this directory data.

Important: If you do not import an LDIF file to populate the LDAP database, you must complete the following steps:

1. Create an entry for the suffix as an organization.

   For example, o=IBM,c=US organization. This can be performed using the Directory Management Tool (see 17.8, “SecureWay Directory Management Tool (DMT)” on page 516).

2. Create an entry for the organization unit

   For example, ou=austin,o=ibm,c=us

3. To view the entries, browse the directory tree from the DMT, where you will see a hierarchy like the one shown in Figure 17-12 on page 501.

Failure to create an entry for this suffix will result in the SecureWay Directory and WCS not working properly.

4. Restart the SecureWay Directory Server

   From the left-hand frame of the Directory Server Administration GUI, click the upper right icon next to the question sign.

Verify the data entries

There are several ways to verify the data entries.

- Scan the error log

  Scanning the error log is a good way to verify whether or not an update has been successful. Remember that an LDIF file may contain a large number of entries, and errors may occur on just single entries of such a bulk load or update. For this reason, you should always check the error log.

- Directory Management Tool

  To view data entries using the DMT, refer to 17.8, “SecureWay Directory Management Tool (DMT)” on page 516.

- SecureWay Directory Server Administration GUI

  a. Log on to the SecureWay Directory Server Administration GUI `http://<fully_qualified_host_name>/ldap`.

  b. Click **Directories/Access control** -> **Browse tree**.

  c. Select and expand the suffix you want to display.

  d. Select and expand the top directory you want to display.

  e. Click any suffix and a new window will appear showing the attributes for that suffix.

  If the update was successful the directory entries will look similar to what is displayed in Figure 17-12, and Figure 17-13.

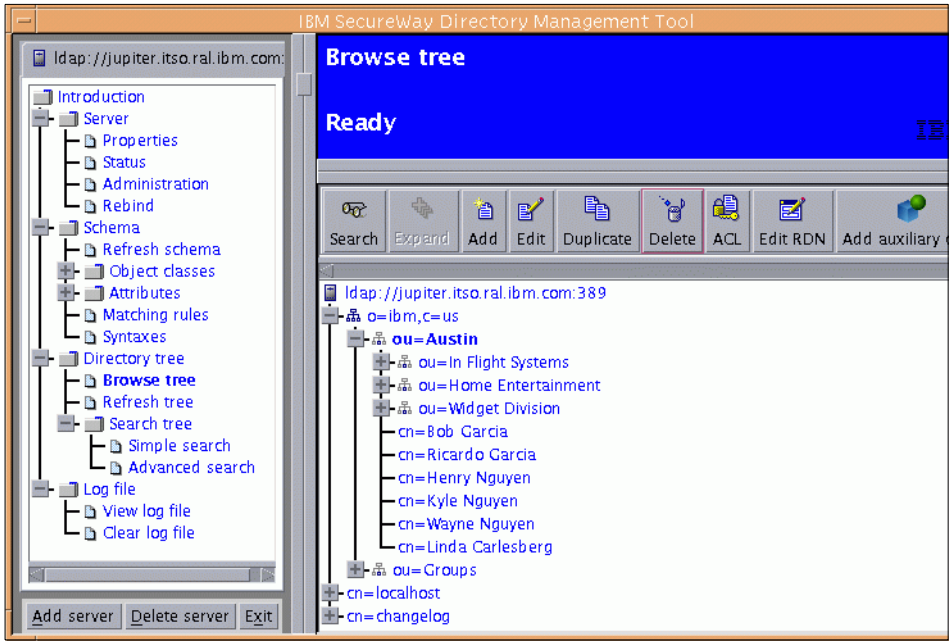


Figure 17-12 Browse directory tree - using Web-based GUI

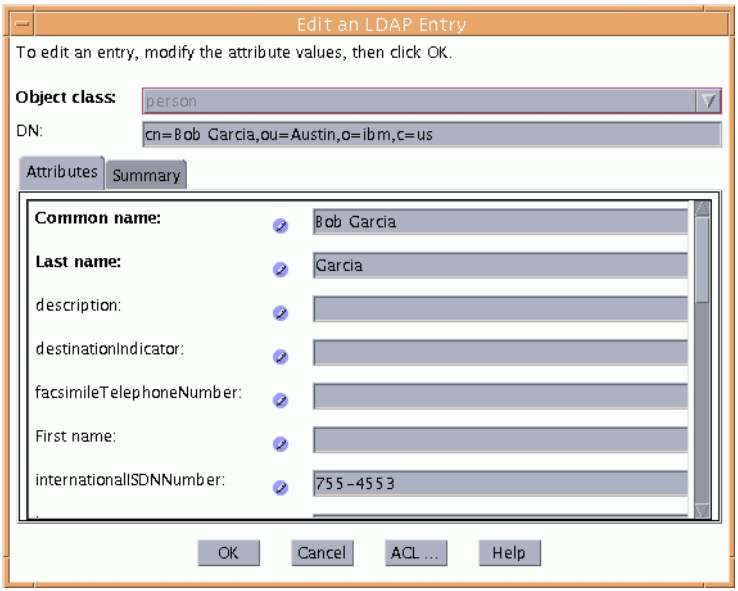


Figure 17-13 Browse entry's attributes and values - using Web-based GUI

17.6 SecureWay Directory V3.2 SSL implementation

IBM SecureWay Directory V3.21 provides Secure Sockets Layer (SSL) Version 3 support, both for the directory server and client. SSL provides encryption of data and authentication using X.509v3 public-key certificates. The directory may be configured to run with or without SSL support. IBM SecureWay Directory V3.2 also supports LDAP referrals, allowing directory operations to be redirected to another LDAP directory server. Replication of the LDAP Directory is supported and allows for additional copies of the directory to be available for directory read operations, which increases performance and reliability when accessing directory information.

The SecureWay Directory V3.21 alone does not provide the capability for SSL connections from LDAP clients. The SSL feature is added by installing the IBM GSKit 4.0.3.X package. The GSKit package includes Secure Socket Layer (SSL) support and associated RSA (3) technology. There are two GSKit packages available: US and Export. They come with different encryption strengths for pre-existing applications. Either package provides 56-bit encryption for new and existing applications.

Note: This redbook covers the 128-bit encryption strength only.

SSL Support (128-bit encryption) / SSL GSKit

The IBM Global Security Kit, Version 4.0.3.X (GSKit) is an optional software package that is required only if Secure Sockets Layer (SSL) security is to be used. For SSL capability, it is necessary to install the IBM GSKit package. A version of GSKit 4.0.3.X is packaged with the SecureWay Directory V3.21 in a US domestic package and an export package. When using SecureWay Directory with WebSphere Commerce Suite, we recommend that you configure SSL (Secure Sockets Layer) in order to encrypt the communication between SecureWay Directory Server and WebSphere Commerce Server.

Important: You should have successfully verified that LDAP is working properly before proceeding with implementing SSL support. If not, it is highly recommended that you return to the installation prerequisites and review the status of your current installation (see Table 17-7).

Table 17-7 Check list

Target system	Installation step	Description	Result / Verification
AIX - LDAP server	Install GSKit for 128-bit encryption	Enables high encryption for LDAP	Check installed file sets via smitty
AIX - LDAP server	Install SSL support for IHS 1.3.12.0	Enables high encryption for IHS	Check installed file sets via smitty and try to request `http://<your_hostname>/back.gif`
AIX - LDAP server	Configure SecureWay for SSL	Mandatory step	Check installed file sets via smitty; in a terminal window type: `java -fullversion`
AIX - LDAP server	Restart LDAP server	Mandatory step	Check installed file sets via smitty and try to request `http://<your_hostname>/back.gif`

Note: Although SSL support is optional, we strongly recommend that you install your server with SSL support. Without SSL-enabled user IDs, passwords will be passed from the WebSphere Commerce Suite server to the SecureWay Directory server without encryption.

17.6.1 Prerequisites

- GSKit - SSL installation
- IBM HTTP Server - SSL configuration

The IBM HTTP Server must be configured for SSL prior to the SecureWay Directory SSL configuration.

17.6.2 GSKit installation

Global Security Kit (GSKit) is an optional software package that is required when using Secure Socket Layer (SSL).

SecureWay Directory V3.21 alone does not provide the capability for SSL connections from SecureWay Directory clients. The SSL feature is added by installing the IBM GSKit package (located on the AIX Bonus Pack). The GSKit package includes SSL support and the associated RSA (4) tech pre-reqs (ldap.max_crypto_client).

gskru301 package

The gskru301 GSKit package is required to install the SSL support for use by the SecureWay Directory client and server (up to 128-bit and 3DES encryption). You should also install the ldap.max_crypto_server or ldap.max_crypto_client to upgrade the SecureWay Directory utilities and the Java Naming and Directory Interface (JNDI) support to use the higher levels of encryption. If you do not install the ldap.max_crypto_server or ldap.max_crypto_client, and only install the gskru301 package, then the SecureWay Directory file sets installed from the base operating system CD provide a maximum of 56-bit DES encryption for SSL.

Although SSL support is optional, we strongly recommend that you install your server with SSL support. Without SSL-enabled user IDs, passwords will be passed from the WebSphere Commerce Suite Server to the SecureWay Directory Server.

gskre301 package

The gskre301 GSKit package is required to install SSL support for use by the SecureWay Directory client and server (up to 56-bit DES encryption). Install the ldap.exp_crypto_client only if you have Java applications that use the JNDI interface with SSL. By installing the gskre301 package, the SecureWay Directory file sets installed from the base operating system CD are enabled to support a maximum of 56-bit DES encryption for SSL.

To install a secure SecureWay Directory from the SecureWay Directory package, first install the client or server from the package. Then the GSKit 3.01 can be installed from the AIX Bonus Pack.

The SecureWay Directory server works without the GSKit installed. In this case it accepts only non-SSL connections from SecureWay Directory clients. Similarly, the SecureWay Directory client works without the GSKit installed.

Make sure the file sets in Table 17-8 exist in the appropriate versions on your LDAP server machine:

Table 17-8 LDAP file set versions

Filename	Version	Description	Where to obtain
gskit.rte	4.0.3.61(or higher)	AIX Certificate and SSL Base Runtime	Included in SecureWay V3.21 package or AIX Bonus Pack
gskrf301.base or	3.0.1.84 (or higher)	Basic encryption module	Included in SecureWay V3.21 package or AIX Bonus Pack
gskru301.base	3.0.1.84 (or higher)	High encryption module	Included in SecureWay V3.21 package or AIX Bonus Pack

Note: Ensure that SSL is configured for the IBM HTTP Server V1.3.12 (see 17.3.8, “IBM HTTP Server V1.3.12 installation” on page 488.

17.6.3 SecureWay Directory SSL configuration

To configure SecureWay Directory SSL, complete the following steps:

1. Start a Web browser, enter URL: `http://<full_qualified_host_name>/ldap`
2. Log on to IBM SecureWay and enter the following:
   - User ID: `cn=root` (distinguished name)
   - Password: `<your_password>`
   - Click **Logon**
3. From the left frame, select **Security** -> **SSL** -> **Settings**
4. When the Server SSL (Secure Sockets Layer) properties window appears, enter the required fields as seen in Figure 17-4, then click **Update**.

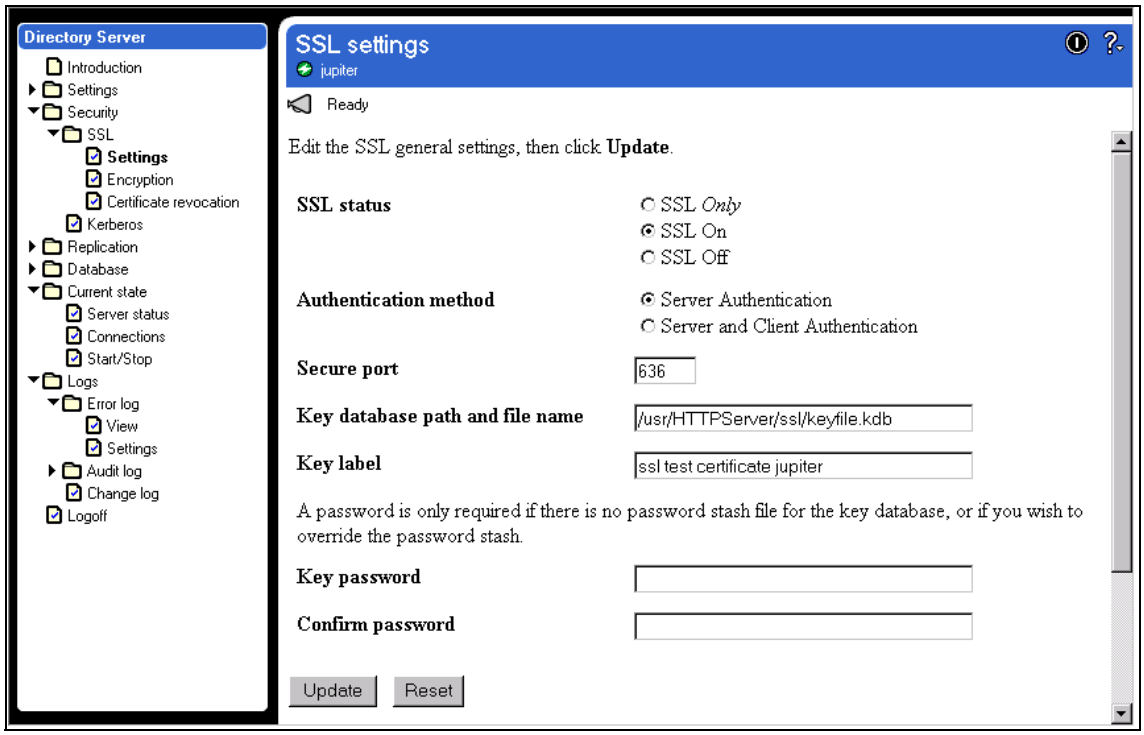


Figure 17-14 SecureWay Directory V3.2 SSL configuration settings

Important: The key database path, file name, key label, and key password must match the configured values of the IBM HTTP Server. Refer to “Configure SSL for the IBM HTTP Server” on page 149 for detailed information.

5. In order to use 128-bit encryption strength you *must* additionally configure SecureWay Directory V3.2 and set the default encryption level to 40-bit encryption strength only as seen in Figure 17-15.

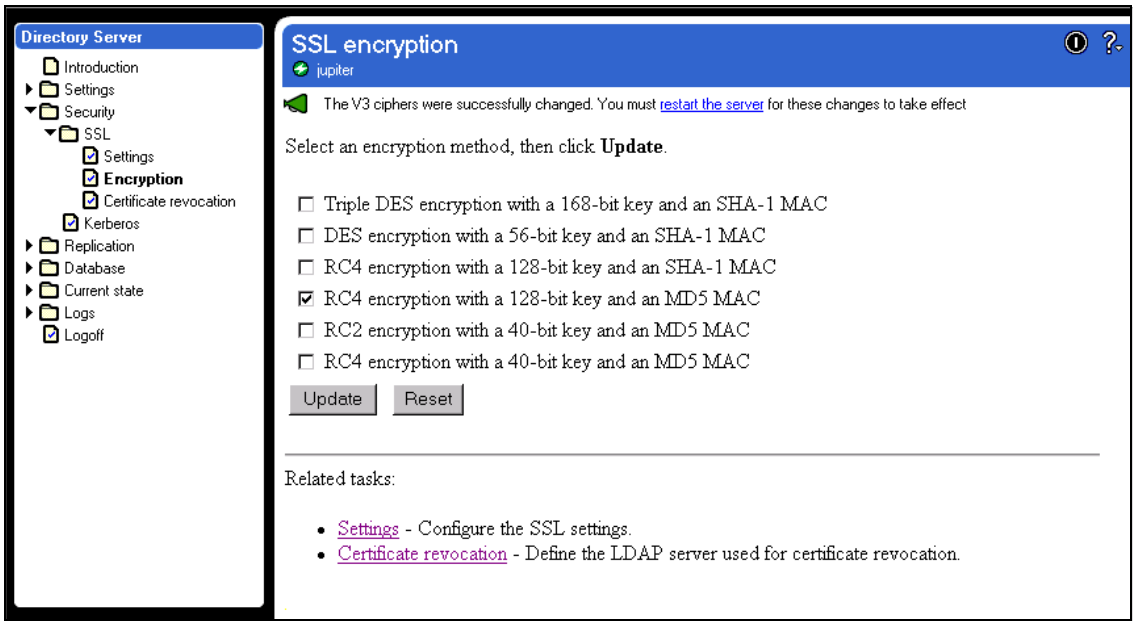


Figure 17-15 SSL encryption configuration window

6. Restart the SecureWay Directory Server
   - Click the icon next to the question sign located in the upper-right corner of the SSL settings window.
   - Wait for the confirmation message or check the server status in the left-hand frame.
7. The SecureWay Directory Server can now communicate in both SSL and non-SSL modes with the WebSphere Commerce Server.
8. Select the new directory you just created, and click the **Search for** button in the upper-right corner.
9. When the Basic Search window appears, enter * in the name field, and click **Search**, as shown in Figure 17-16.

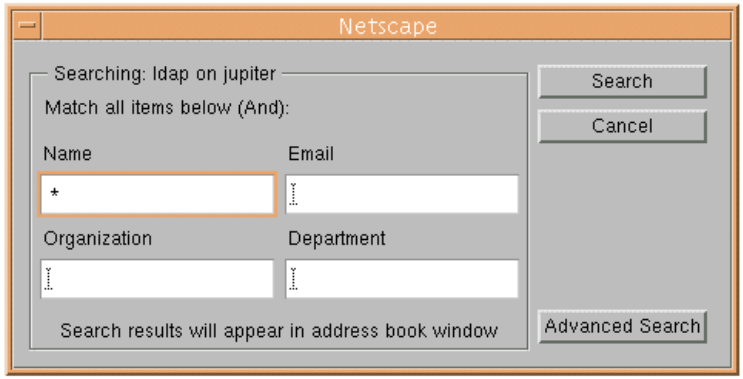


Figure 17-16 Netscape address book basic search window

10. A list showing all users, like the one seen in Figure 17-17, should be visible in the right-hand frame.

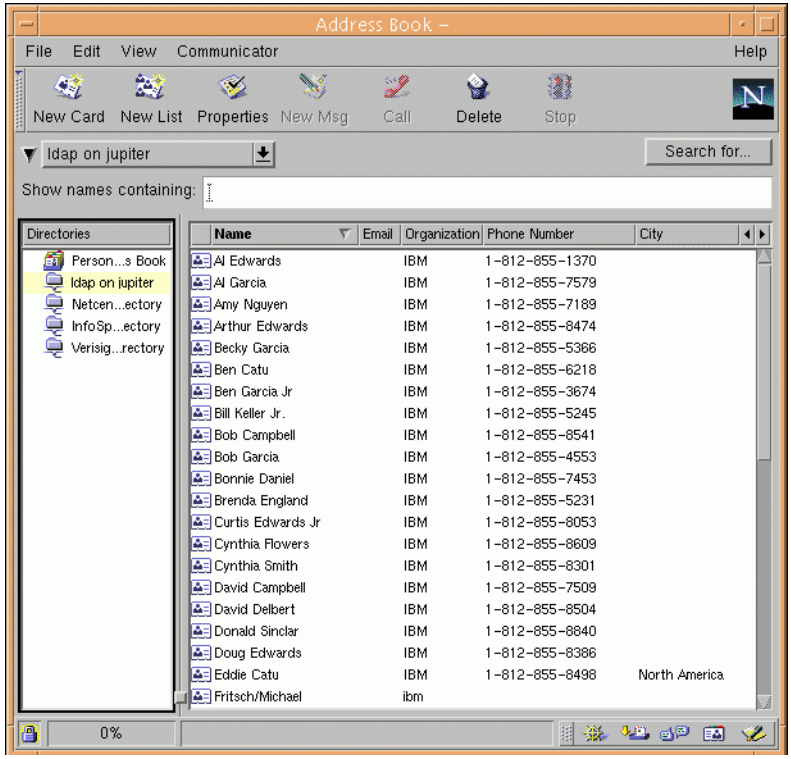


Figure 17-17 Netscape Address Book search result

Note: It is crucial to restart your LDAP server after updating the SSL settings.

17.7 Configuring WCS V5.1 for LDAP

To configure WebSphere Commerce Suite V5.1 to work with SecureWay Directory V3.2, complete the following steps:

1. Install and configure the SecureWay Directory V3.2 server as described in 17.4, “SecureWay Directory V3.2 configuration” on page 491, and in accordance with the *IBM SecureWay Directory V3.2 for AIX Installation and Configuration Guide*.

   Install the directory server on a machine that is behind the same firewall as your Commerce Suite machine.

2. On your WCS machine, configure WCS to work with SecureWay Directory V3.2 using the WCS Configuration Manager, as described in 7.6.1, “Creating a WCS instance using the Configuration Manager” on page 166.

3. Expand **Configuration Manager** -> **node** -> **instance** -> **<your_instance>** -> **instance properties** -> **Member SubSystem**.

4. Use the Member SubSystem window of the Configuration Manager to configure Commerce Suite to use a directory server by selecting **LDAP** as the mode of authentication. If you select LDAP, the rest of the fields on this panel will be enabled.

   Table 17-9 shows all field descriptions and the available options.

Table 17-9 WCS Configuration Manager - member subsystem panel

Label	Explanation
LDAP Version	The version of the LDAP protocol that the WebSphere Commerce Server will use to communicate with the LDAP server. The default is V3. Choose **V3**.
LDAP Type	Select the Directory Server software you are using with Commerce Suite. Choose **SecureWay Directory Server**.
Single Sign-on	Select this check box to allow users who have already been authenticated by WebSphere Application Server to be recognized by Commerce Suite. At the time of writing, this feature was not enabled (grayed out).

Label	Explanation
Host	The fully qualified host name specifying where the LDAP server is installed.
Port	The port used by the LDAP server. The default port for non-SSL connections is 389. If you want to use SSL, the default port to use with SecureWay is 636.
Administrator DN	The distinguished name of the LDAP server administrator as entered during SecureWay V3.21 configuration.
Administrator Password	The LDAP server administrator's password.
Confirm Password	Re-enter the LDAP administrator's password.
LDAP Authentication Mode	Specifies the authentication mechanism that the LDAP server uses: ► None means that Commerce Suite does not authenticate to the LDAP server. ► Simple means that Commerce Suite uses a distinguished name and password to authenticate to the LDAP server.
Time out	The time in seconds before an LDAP search times out.
Mapping File Name	The XML file that contains information on which Commerce Suite access bean attribute maps to which LDAP attribute. Default is set to ldapmap.xml.
LDAP search root	The distinguished name of all the search roots, separated by semicolons.
Shopper Base Distinguished Name	The distinguished name of LDAP entry under which user entries for new users registered from WebSphere Commerce Suite will be stored.
Shopper Objectclass	The objectclasses used to create user entries on the LDAP server, separated by spaces.
Organization Objectclass	Reserved for future use.
Organization Unit Objectclass	Reserved for future use.

Label	Explanation
Person RDN	The LDAP attribute to be used as the RDN attribute (also known as the naming attribute) when a user entry is created in LDAP. If Basic authentication mode is selected in the Web server window of the Configuration Manager, Commerce Suite uses the Commerce Suite user logon ID as the RDN attribute value. If X.509 authentication mode is selected, Commerce Suite uses the Subject name from the user's X.509 certificate as the RDN attribute value.

5. Modify your ldapmap.xml file as needed. This file specifies the mapping between Commerce Suite attributes and LDAP attributes. Data is replicated between Commerce Suite and LDAP based on the content of this file.

   Note: For more details on the syntax of the ldapmap.xml file and how to modify its contents, see the WebSphere Commerce Suite online help on LDAP.

6. If you need to extend your database schema programmatically, see the WCS online help on LDAP.
7. To make your changes become effective, click **Apply**.
8. We recommend to restart your WebSphere Commerce Server - <wcs_instance>.

Note: Make sure you clean up WCS, WAS and your Web browser's cache before restart the WebSphere Commerce Server - <wcs_instance> from the WebSphere Administrative Console.

After completing the configuration, the Configuration Manager window should look like Figure 17-18.

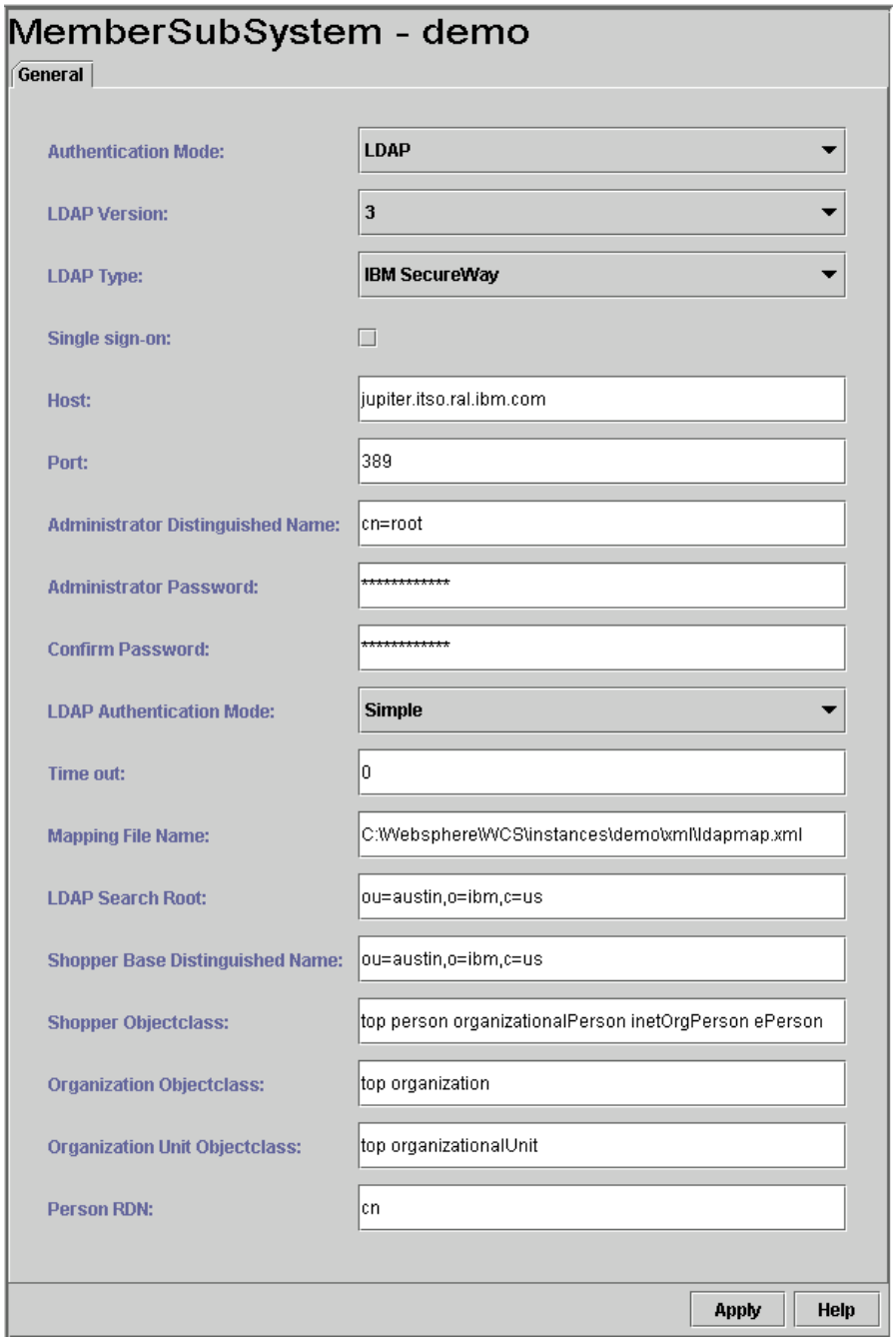


Figure 17-18 Configuration Manager LDAP configuration

Note: We have verified the SecureWay Directory V3.2 installation to work with a WCS 3-tier runtime environment for AIX.

Make sure you supply the appropriate path in the mapping the file name field. For example:

```
/usr/lpp/CommerceSuite/instance/<wcs_instance>/xml/ldapmap.xml
```

17.7.1 WCS SSL support

In order to make WCS use SSL connections for communication to SecureWay Directory V3.2, make sure you change the port number from 389 (LDAP) to 636 (SSL-LDAP) in the WCS Configuration Manager - member subsystem panel.

17.7.2 Configuration verification

There are several checkpoints to verify the SecureWay Directory V3.2 configuration:

- Check LDAPDB2 database
- Start and stop LDAP server
- Check the error logs
- Import sample data ldif files

17.7.3 IBM SecureWay Directory V3.21 SSL verification

If unexpected errors occur in the background while implementing the 128-bit encryption strength, you may still be able to use LDAP support with certain encryption, for example, only 56-bit available instead of the desired 128-bit encryption strength.To prove that your system is actually using 128-bit encryption strength, it is crucial to double-check all settings and perform the following security verification “test“.

Note: We performed the following test using Netscape Navigator 4.73i for AIX and V4.75 for Windows NT. Both were English versions and capable of 128-bit encryption strength. Make sure you have these or equivalent versions available, as there may be some issues about other versions.

17.7.4 128-bit encryption security verification

Complete the following steps to verify the 128-bit SSL configuration:

1. Start the Netscape Navigator browser (4.73i for AIX or 4.75 for Windows NT).

2. In the Navigator pull-down menu, click **Address book**.
3. When the Address book window appears, click **File** -> **New Directory**.
4. When the Directory Server property window appears, fill in the required fields as seen in Figure 17-19 and click **OK**.

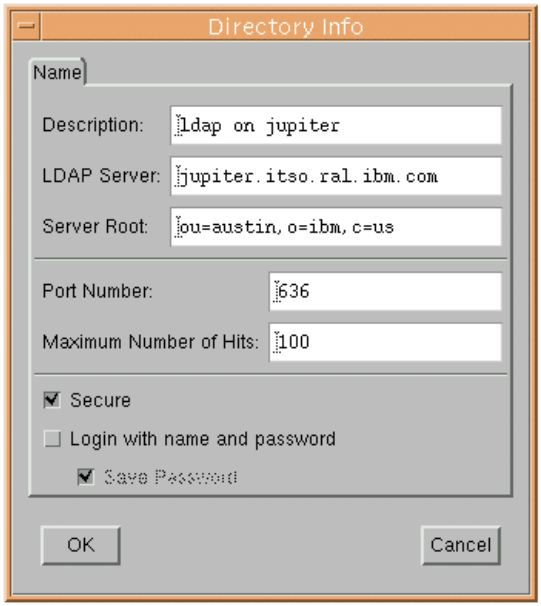


Figure 17-19 Netscape Address Book property window

At this point you can choose either secure or standard connection and thereby verify both settings. The port number will change automatically.

5. Check **Secure** and click **OK**.
6. Click the **Security** button on the Netscape Navigation Toolbar.
7. The Security window pops up. Click **Navigator** on the left-hand side of the window.
8. The window now displays the Navigator security settings. As we are using SSL V3, you should **disable** SSL V2 just for this test.
9. Click the little square in front of the label showing Enable SSL (Secure Sockets Layer) V2. (Windows NT users simply remove the tick in the checkbox by clicking it.)
10. Make sure that SSL V3 is **enabled**.
11. Click **Configure SSL V3** in the lower right of the window.

12. The Netscape: Configure Ciphers window pops up. As we have set the encryption method in the SSL encryption window of the SecureWay configuration GUI to RC4 encryption with a 128-bit key and an MD5 MAC, we perform this part of the test with this option not selected.
13. Deselect the following ciphers:
    - RC4 encryption with a 128-bit key and an MD5 MAC
    - No encryption with an MD5 MAC
14. Select **OK** in this and the following window.

Note: We recommend that you restart your browser to make the changes become effective.

15. When you try to access the LDAP address book now or try to perform a search as shown above, you will get a Netscape: Error pop-up window. Click **OK** to close the pop-up window.
16. Now open the Netscape: Configure Ciphers window again. Select the following cipher, and deselect all other options:

    ```
    RC4 encryption with a 128-bit key and an MD5 MAC
    ```

17. Select **OK** in this and the following window.

Note: We recommend that you restart your browser to make the changes become effective.

18. When you try to access the LDAP address book now or try to perform a search as shown above, a list showing all users should be visible in the right-hand frame (see Figure 17-20).

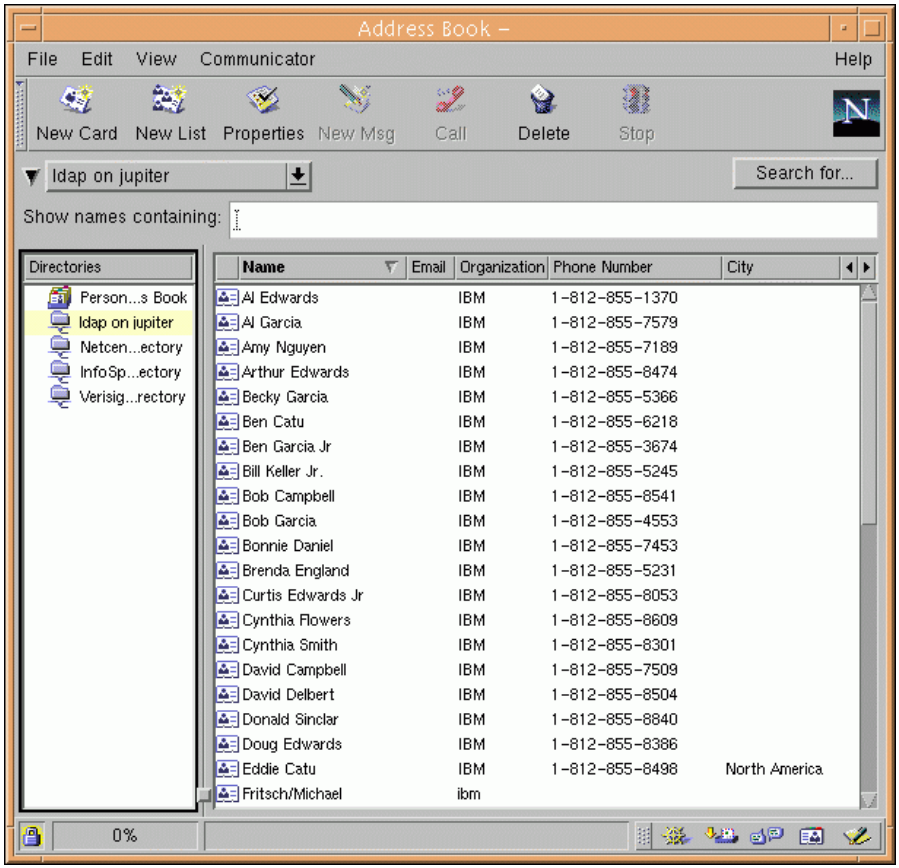


Figure 17-20 Netscape address book search list of users

This concludes the 128-bit encryption verification. We have successfully installed and configured the IBM SecureWay Directory Server V3.21 with SSL 128-bit encryption strength ready for use with WebSphere Commerce Suite V5.1.

Note: You may want to undo the changes made for this verification test in Netscape as well as in SecureWay Directory V3.2.

17.8 SecureWay Directory Management Tool (DMT)

The IBM SecureWay Directory Management Tool (DMT) provides a graphical user interface that enables you to manage information stored in directory servers.

The tool can be used for the following tasks:

- Connecting to directory servers
- Displaying server properties
- Administering schema object classes and attributes
- Administering directory entries
- Administering directory entry ACLs

LDAP is a client/server protocol for accessing a directory service. It was initially used as a front-end to X.500, but can also be used with stand-alone and other kinds of directory servers. LDAP can be used as a centralized information repository to support information sharing among various clients.

17.8.1 Connecting to directory servers

This section provides detailed instructions for connecting to SecureWay Directory servers.

Server log on

If a directory user DN and password are not provided in the DMT configuration file, the tool connects as an anonymous user once it is started. Although an anonymous user can browse the directory tree and schema, you need to log on as a directory user to perform directory updates, in most instances. To modify the directory server schema you must log on as the server administrator.

To log on as a different user complete the following steps:

1. Start the Directory Management Tool (DMT):
   a. Move to the SecureWay Directory Server machine to start the DMT.
   b. Log in to AIX as user root and start an AIX terminal session.
   c. Start the DMT by typing the following:

      ```
      # dmt
      ```
2. Click **Server** -> **Rebind**.
3. Provide the user DN and password, and click Enter.

The DMT uses the /usr/ldap/etc/dmt.conf configuration file. A sample dmt.conf file follows:

```
#browser=server1.url=ldap://jupiter.itso.ral.ibm.com:389
#server1.security.bindDN=
#server1.security.password=
#server1.security.ssl.keyclass=
#server1.security.ssl.keyclass.password=
```

In this file you can specify LDAP servers URL, port, user DN, password, keyclass file name, and password for SSL connections.

Important: We strongly recommend for security reasons that you do not provide DN user name and password in the dmt.conf file in a production environment.

Connect to an additional server

If localhost is provided in the URL of the configuration file, DMT will connect to the LDAP directory server on the local machine where DMT is running. To add a connection to another directory server, complete the following steps:

1. From the DMT, click **Add server**.
2. Provide the server name, port, user DN, and password.
3. Press Enter.

Note: Do not specify the URL prefix (for example: ldap://) in the server name.

You can choose to connect to the server via SSL by first selecting **Use SSL** and then providing the keyclass file name and password.

Display server properties

You can view the server properties by clicking **Server** -> **Properties**. The current bind DN, subschema entry, supported LDAP protocol versions, and the naming entries that the server holds are displayed.

17.8.2 Administering schema object classes and attributes

This section provides information for browsing, adding, editing, and deleting schema attributes and object classes.

Browse directory schema

IBM SecureWay Directory V3.21 provides dynamically extensible schema support. An administrator can define new attributes and object classes to enhance the default schema. The directory schema can be browsed and updated with the DMT. You must log on as a directory administrator to update the schema.

To browse the directory schema complete the following steps:

1. From the DMT, click **Schema** -> **Browse schema**.
2. Click the **+** sign to display the following selections:

- Attribute types

  View all defined attributes, or alternatively click **Schema** -> **Attributes** -> **View attributes**.

- Object classes

  View all defined object classes, or alternatively click **Schema** -> **Object classes** -> **View object classes**

- Syntaxes

  To view all supported syntaxes.

- Matching Rules

  View all supported matching rules.

Add an object class

To add an object class, complete the following steps:

1. From the DMT, click **Schema** -> **Object classes** -> **Add object class**.
2. Provide the object class name, description, and a unique string of object identifier (the OID).
3. Select the superior object class from which to inherit attributes.
4. Determine the object class type (the default is structural).
5. Click the **Required attributes** tab:
   - Select **MUST have attributes** from the attribute list in the left frame.
   - Click **Add** to move the selection to the right frame.
   - You can also select the attributes in the right frame and then click **Remove** to deselect them.
   - Click **OK**.
6. Click the **Optional attributes** tab, and then select **MAY have attributes**.
7. Click **OK**.

Edit an object class

This operation is similar to adding an object class except that a pull-down menu is provided for the selection of the object class to be edited. To edit an existing object class, complete the following steps:

1. Click **Schema** -> **Object classes** -> **Edit object classes**.
2. Select an object class to be edited from the pull-down menu.
3. Provide the description.
4. Click the **Required attributes** tab and select **MUST have attributes**.

5. Click **OK**.
6. Click the **Optional attributes** tab and then select **MAY have attributes**.
7. Click **OK**.

Delete object classes

To delete one or more object classes complete the following steps:

1. Click **Schema** -> **Object classes** -> **Delete object classes**.
2. Select the object class or classes to be deleted.
3. Click **Delete**.
4. Click **OK** to confirm the deletion.

Add an attribute

To add an attribute complete the following steps:

1. Click **Schema** -> **Attributes** -> **Add attributes**.
2. Provide an attribute name, description, and an OID.
3. Select a syntax for this attribute from the list.
4. Determine whether this is a multi-valued attribute.
5. Select the matching rules used.
6. Click **OK**.

Note: Advanced users can click the IBM extensions tab to change the DB2 table name, the DB2 column name, the security class, and the indexing. We recommend indexing an attribute to optimize performance for LDAP searches.

Edit an attribute

This operation is similar to adding an attribute process except that a pull-down menu is provided for the selection of the attribute to be edited. To edit an existing attribute, complete the following steps:

1. Click **Schema** -> **Attributes** -> **Edit attributes**.
2. Select an attribute from the list.
3. Make the necessary entries in the General tab, then click **OK**.

Note: Advanced users can click the I**BM extensions** tab to change the DB2 table name, the DB2 column name, the security class, and indexing. We recommend indexing an attribute to optimize performance for LDAP searches.

To delete one or more object classes complete the following steps:

1. Click **Schema** -> **Attributes** -> **Delete attributes**.
2. Select the attribute or attributes to be deleted.
3. Click **Delete**.
4. Click **OK** to confirm the deletion.

17.8.3 Administering a directory tree

This section provides information for browsing, searching, adding, editing, and deleting schema attributes and object classes.

Browsing a directory tree

You can browse the directory tree by using the Browse tree option. When you browse the directory tree, the directory contents are displayed according to the directory hierarchies. To open part of the tree, expand the entries. The entries that are in the next level down are displayed. To browse the directory tree, complete the following steps:

1. Click **Tree** -> **Browse tree**.
2. To expand the tree one level, click a **+** sign.

   The tool bar at the top of the window allows for an operation on a selected entry in the tree to be initiated. The operations include Add, Search, Edit, Delete, Expand, ACL settings, and Edit RDN. Use **Edit** to view an entry. When an entry (a node in the tree) is selected, click the **Expand** button to expand the entire subtree below the entry.

   Double-click an entry to edit it.

Searching a directory tree

To allow convenient access to entries of special object classes such as user and group, DMT provides a simple search option in addition to a full-search option. While the full-search option allows you to provide complete specifications of all parameters to a directory search operation, the simple search requires only minimal input for searching through a set of entries that belong to a selected object class.

Simple search

To perform a simple search, complete the following steps:

1. Click **Tree** -> **Search tree** -> **Simple search**.
2. Select the type of entry to search.
3. Determine the filter for the search result.

4. Click **OK**.

Simple search

To perform a simple search, complete the following steps:

1. Click **Tree** -> **Search tree** -> **Simple search**.
2. Select the type of entry to search.
3. Determine the filter for the search result.
4. Click **OK**.

Full search

To perform a full search, complete the following steps:

1. Click **Tree** -> **Search tree** -> **Full search**.
2. Input the search constraint:
   - Search base DN (the default is all suffixes)
   - Scope (the default is subtree)
   - Size limit (the default is unlimited)
   - Time limit (the default is unlimited)
   - Alias de-referencing (the default is no)
   - Referral chasing (the default is yes)
3. Click the **Search filter** tab at the top of the display.
4. Input the search filter. If necessary, use the AND or OR connectors.
5. Click the **Search return set** tab.
6. Select the attributes to be returned or the full entry.
7. Click **OK**.

Adding an entry

To add an entry to the directory tree, complete the following steps:

1. Click **Entries** -> **Add entry**. You can also add an entry if you click **Browse tree**, select the parent entry, and then click **Add** on the toolbar.
2. Provide the parent DN and the Relative Distinguished Name (RDN) for the new entry. The RDN must be entered as an attribute=value pair.
3. Choose the object type (object class) from the list or click **Other** for more options. If **Other** is selected, you can specify either a structural object class or an auxiliary object class or both.
4. Click **OK**.

5. Another window displays the attributes associated with the selected object class. Highlighted fields are required fields. Enter the attribute values for the entry. Use the edit icon to add multiple values.

When the action is initiated from the tree browsing pane, the parent entry can be selected from the directory tree and the parent DN is entered automatically.

Note: The flyover that appears when the cursor is positioned over the attribute name or the text field describes the syntax of that attribute.

Edit an entry (or view an entry)

To view an entry, complete the following steps:

1. Click **Entries** -> **Edit entry**. You can also select an entry if you click **Browse tree** and then double-click the entry.
2. Provide the entry DN to edit or view.
3. Click **OK**.
4. Another window displays the attributes associated with the selected object class. Highlighted fields are required fields. Enter the attribute values for the entry. Use the edit icon to add multiple values.

Like adding an entry, this operation can be launched from the browsing tree window. The entry can be edited from the tree by double-clicking it.

Delete an entry

To delete an entry from the directory tree, complete the following steps:

1. Click **Entries** -> **Delete entry**, or from the browsing window click the entry to be deleted.
2. Click **Delete**.
3. Click **OK** to confirm the deletion.

Edit an entry RDN

To edit an entry RDN, complete the following steps:

1. Click **Entries** -> **Edit entry RDN**. You can also select an entry if you click **Browse tree** and then click the entry.
2. Enter the current DN and provide the new RDN for the entry. (You can change the new RDN; the current DN is not editable.)
3. Click **OK**.

17.8.4 Administering directory entry ACLs

To modify an ACL for an entry, complete the following steps:

1. Click **Entries**- > **ACLs**. Enter the DN and then click **OK**. You can also select the entry if you click **Browse tree** and then click the **ACL** button on the top of the window.
2. On the ACLs tab:
   - Either edit the existing list or create a new subject ACL list for a new subject.
   - Determine whether descendant entries are to inherit the ACL lists.
   - Mark the **Remove** check boxes for those subject ACL list that are to be removed.
   - Mark/unmark the check boxes for the rights to add/remove.
   - Mark/unmark the check boxes for the rights to read/write/search/compare the attributes of three security classes.
   - To add a new subject ACL list, enter the subject DN, select the subject type, and then click **ADD**. A new list is added and ready for changes.
3. Click the **Owners** tab:
   - Determine whether descendant entries are to inherit the owner list.
   - Mark the **Remove** check boxes to remove owners.
   - To add a new owner, enter the owner DN, select the owner type, and then click **ADD**.
4. Click **Change**.

17.8.5 Troubleshooting

The first time you edit a suffix or add an entry to a suffix, you may see the following message:

```
An error occurred getting attributes for entry c=us: noSuchObject.
```

This means that the suffix contains no data.

To add data to a suffix

1. In the navigation menu click **Entries** -> **Add entry**.
2. Leave the Parent DN blank and specify the suffix as the entry (for example `c=us`).
3. Select the object type (for example **Country**) and click **OK**.

4. Fill in the desired attribute values and click **Create**.

Rebind a suffix

You should now be able to edit the suffix as well as add entries.You can find the bind DN in either of two ways from the menu area:

1. You can locate it in the Rebind to Server window. Click **Server** -> **Rebind** to display the window. The rebind DN will be displayed in the User DN field. (If the bind DN is anonymous, the **Anonymous** radio button is checked.)
2. Click **Server** -> **Properties**. In the table, under Server attributes, find the bind dn property.

17.9 Working example 1

The following working example is *not* designed to show/implement a fully valid WCS-LDAP authentication scenario. It is intended to demonstrates the “flow” between WCS V5.1 an SecureWay Directory V3.2 and is designed to help you set up your development and test environment and validate your configuration and installation.

17.9.1 Three different scenarios

As already mentioned earlier, a user could be created on the LDAP server by:

- Registering through WCS
- Using the LDAP interface
- Using applications other than WCS connected to the same LDAP server

This section does not address the use of other LDAP authentication applications. We focus on the first two methods in our following working example.

17.9.2 Working example prerequisites

The following working example is based on the accurate installation of IBM SecureWay Directory V3.21 and adoption of the WCS member subsystem as shown above.

Note: During the login process the E-mail address field in the InFashion/WebFashion sample store is actually labelling the LOGONID and maps to the appropriate cn (common name) attribute in the SecureWay directory. For mapping details, refer to the WCS online help.

In order to make the sample store registration and login page reflect the actual content of the JSP, you could modify the demo store registration page (for example, the corresponding property file).

For demonstration purposes, only the US English pages are modified.

Note: Your changes will only become effective after a restart of your WebSphere Commerce Server - <wcs_instance> from the WebSphere Administrator's Console.

17.9.3 User entry is replicated from WCS to LDAP

By following the instructions below you will be able to:

- Create a new user in WCS using the demo store registration page
- Log on to WCS as that new user and thereby authenticate to WCS
- Migrate the user entry "on-the-fly" from Commerce Suite to LDAP automatically.

Note: Future logon attempts will still try to authenticate to LDAP.

Figure 17-21 provides an overview of working example 1.

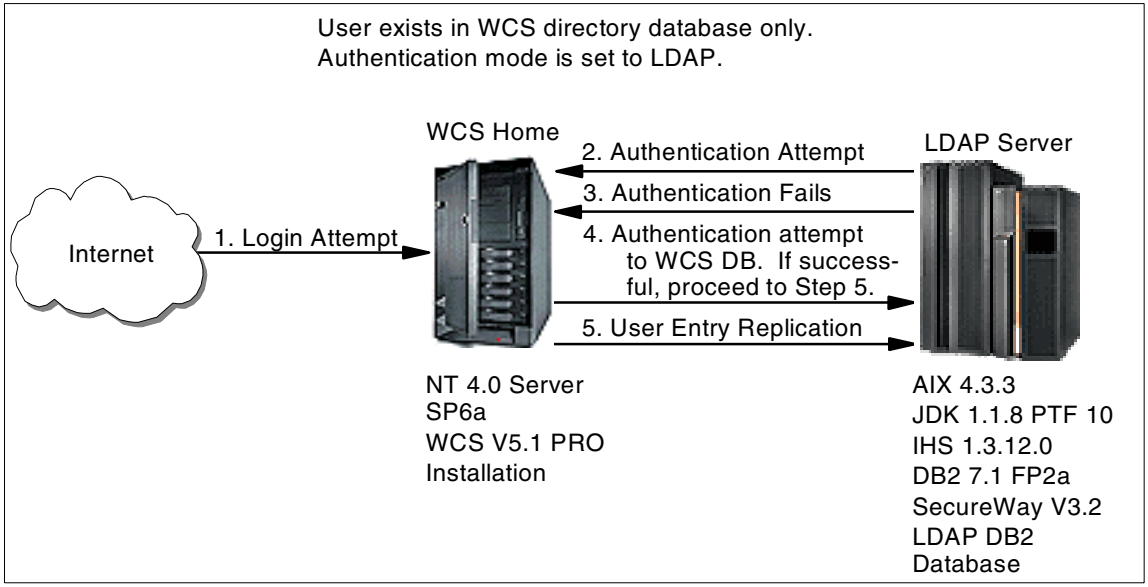


Figure 17-21 Working example 1 overview

Example 1 flow

User entry exists in WCS database only.

1. User attempts to log in.
2. Authentication mode is set to LDAP, so WCS attempts to authenticate the user via LDAP.
3. Authentication fails because the user only exists in the WCS database.
4. WCS attempts to authenticate the user by using the WCS database. If authentication is confirmed, WCS proceeds.
5. The user entry is replicated “on the fly“ to the LDAP database.

The user entry now exists in both databases (LDAP and WCS).

Create a new user in WCS/ Register as a new user

We create a new user in the WCS database by registering in the sample store.

1. Make sure your Commerce Server is running.
2. Start the sample store.
3. Click **Register** in the left-hand frame.

Note: During the login process the E-mail address field in the WebFashion sample store is actually labelling the LOGONID and maps to the appropriate cn (common name) attribute in the SecureWay Directory V3.2.

4. The Register or Login page appears. Under NEW CUSTOMER, click **Register**. Fill in the entry fields. For the password use `test` in lowercase.
5. Keep in mind that the entry in the E-mail address field will become the future LOGONID. Click **Submit**.

   The My account page should be displayed.

You have successfully created a new WCS user and the entry in the WCS database.

To verify if the new user has been created in your WCS database, check the entries in the USERREG table in your WCS instance database.

Migrate the user entry “on-the-fly” from WCS to LDAP

When you register as we just did, the user entry data is replicated “on the fly“ from the WSC database to the LDAP database automatically. No further action is needed.

Restriction: This part of the example requires that you did not restrict the information flow via the ldapmap.xml file. For further information about the ldapmap.xml file, refer to the WCS online help.

To verify if the new user has been successfully replicated to your LDAP database, you can browse the directory tree using the DMT tool on your AIX LDAP server. For details on how to use the DMT tool, refer to 17.8, “SecureWay Directory Management Tool (DMT)” on page 516.

Note: To show the changes we just made, make sure you reload the server schema by refreshing the directory tree if the DMT tool is already running.

For details about what data can be replicated, refer to 17.2.3, “What’s new in WCS V5.1 LDAP support” on page 476.

17.10 Working example 2

The following working example is *not* designed to show/implement a fully valid WCS-LDAP authentication scenario. It is intended to demonstrates the “flow” between WCS V5.1 and SecureWay Directory V3.2 and is designed to help you set up your development and test environment and validate your configuration and installation.

17.10.1 Three different scenarios

As already mentioned earlier, a user could be created on the LDAP server by:

- Registering through WCS
- Using the LDAP interface
- Using applications other than WCS connected to the same LDAP server

This section does not address the use of other LDAP authentication applications. We focus on the first two methods in our following working example.

17.10.2 Working example prerequisites

The following working example bases on the accurate installation of IBM SecureWay Directory V3.21 and adoption of the WCS member subsystem as shown above.

Note: During the login process the E-mail address field in the InFashion/WebFashion sample store is actually labelling the LOGONID and maps to the appropriate cn (common name) attribute in the SecureWay directory. For mapping details, refer to the WCS online help.

In order to make the sample store registration and login page reflect the actual content of the JSP, you could modify the demo store registration page (that is, the corresponding property file).

For demonstration purposes, only the US English pages are modified.

Note: Your changes will only become effective after a restart of the WebSphere Commerce Server - <wcs_instance> from the WebSphere Administrator's Console.

17.10.3 User entry is replicated from LDAP to WCS

By following the instructions below you will be able to:

- Create a new user in IBM SecureWay using the DMT tool on the AIX machine where SecureWay resides.
- Log on to WCS as that new user and then authenticate to SecureWay.
- Migrate the user entry "on-the-fly" from LDAP to Commerce Suite automatically.

Note: Future logon attempts will still try to authenticate to LDAP.

Figure 17-22 provides an overview of this working example.

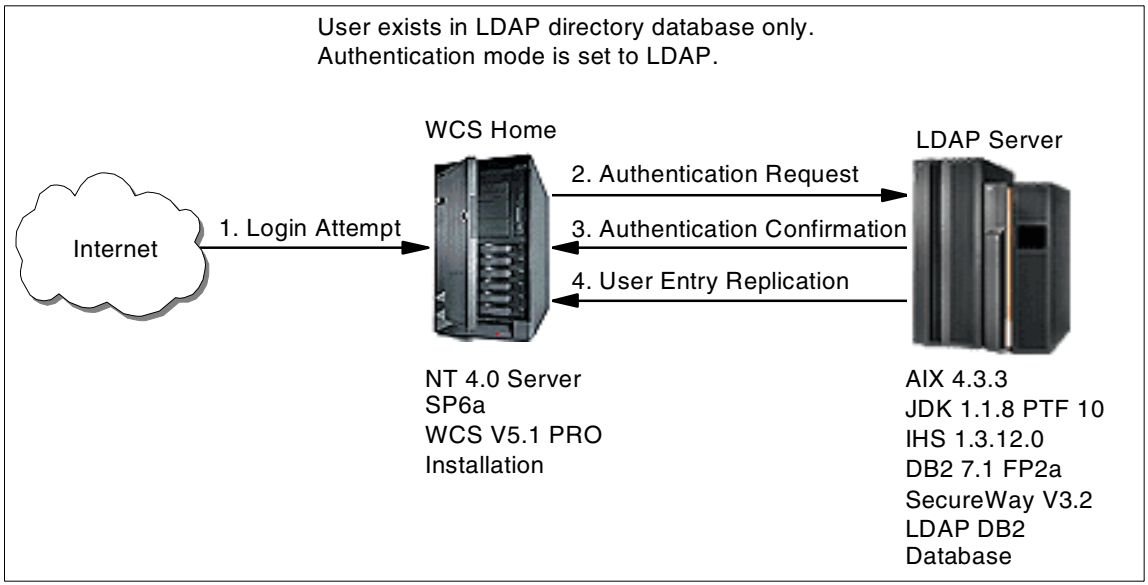


Figure 17-22 Working example 2 overview

Example 2 flow

User entry exists in LDAP database only.

1. User tries to log in.
2. Authentication mode is set to LDAP, so WCS attempts to authenticate the user via LDAP.
3. Authentication is confirmed.
4. The user entry is replicated “on the fly“ to the WCS database.

The user entry now exists in both databases (LDAP and WCS).

Create a new user in SecureWay Directory V3.2

In order to create a new user under the DN entries we created during importing the sample LDIF file, perform the following:

1. Open a terminal window on your AIX system
2. Launch the DMT tool by typing:

   ```
   # dmt
   ```

3. The IBM SecureWay Directory Management Tool Window opens. Expand the Server tree menu in the left-hand frame and click **Rebind.**

4. The Rebind window is shown in the right-hand frame. Click the **Authenticated** button and enter your user DN and user password and then click **OK**.

   For example:

   – `cn=root`

   – `<cn=root's_password>`

5. Expand the directory tree and click **Browse tree**.
6. Your directory structure is shown in the right-hand frame. Expand the o=ibm, c=us tree.
7. Click **ou=Austin**. Your window should look like Figure 17-23.

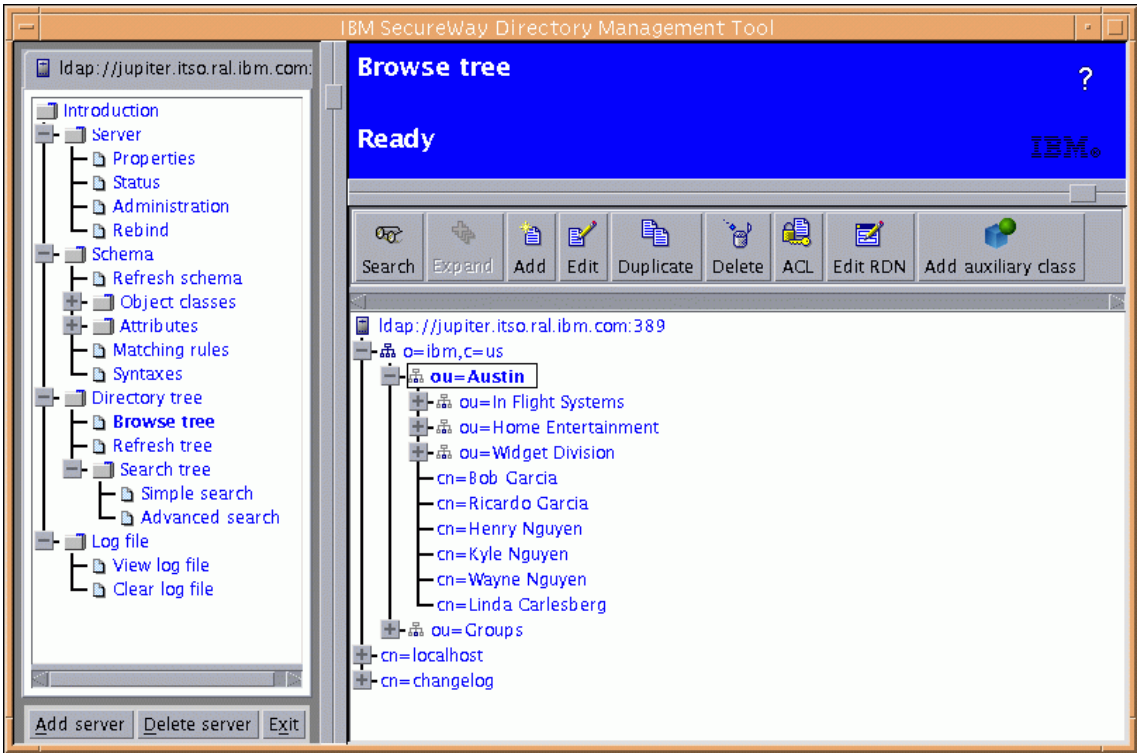


Figure 17-23 Create a new user in SecureWay Directory V3.2

8. Click **Add**. The Add an LDAP Entry window pops up. Do the following as seen in Figure 17-24:

   – Select **User** for Entry Type

   – Parent DN: `ou=Austin,o=ibm,c=us`

– RDN: `cn=Michael Fritsch`

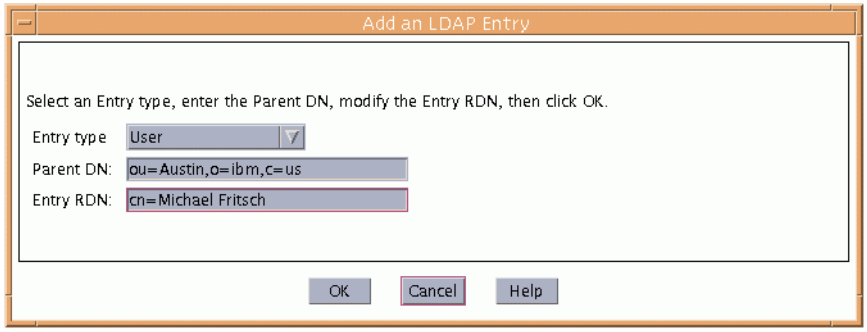


Figure 17-24 Add an LDAP entry

9. Click **OK**. The new user is being created.
10. You must now specify a password for the user by double-clicking the new user entry. Your window should look like the one shown in Figure 17-25.

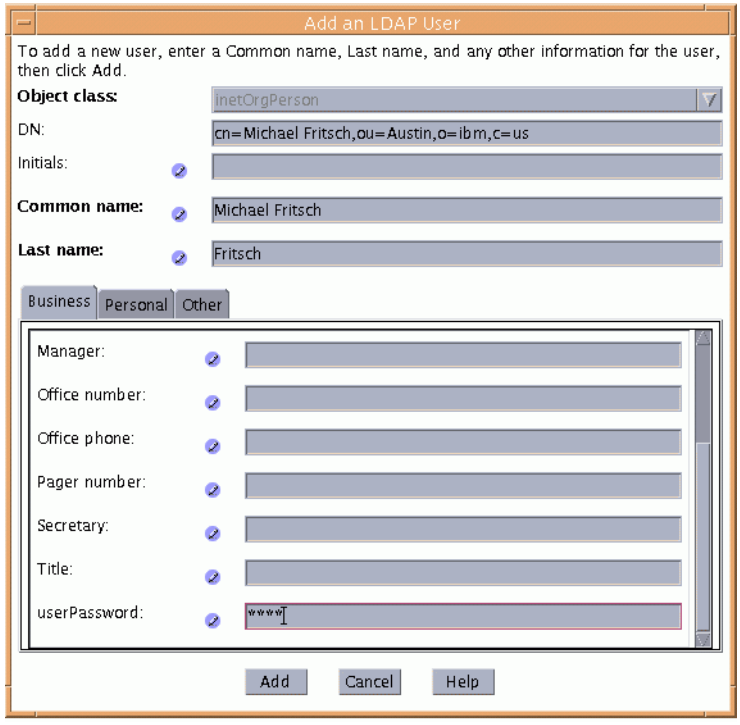


Figure 17-25 Add a password

11. Scroll down until you see the user password entry field. Enter `test` in lowercase.
12. Click **Add**.

This concludes the LDAP part of the user creation process. You have successfully created a new user in the LDAP directory (database).

Log on to WCS

Make sure your Commerce Server is running.

1. Open the sample store and click **Register** in the left-hand frame.

   During the login process the E-mail address field in the InFashion/WebFashion sample store is actually labelling the LOGONID and maps to the appropriate cn (common name) attribute in the SecureWay directory. For mapping details, refer to the WCS online help.

2. The Register or Login page appears. Under RETURNING CUSTOMER, enter `Michael Fritsch` in the E-mail address field, and `test` in the Password field, and then click **Login**.

   The My account page should be displayed.

You have successfully logged on to WCS as a user in the LDAP directory (database).

Migrate the user entry “on-the-fly” from LDAP to WCS

When you log on as the user we just created, the user entry data is replicated “on the fly“ from the LDAP database to the WCS database automatically. No further action is needed.

Restriction: This part of the example requires that you did not restrict the information flow via the ldapmap.xml file. For further information about the ldapmap.xml file, refer to the WCS online help.

To verify if the new user has been successfully replicated to your WCS database, check the entries in the USERREG table in your WCS instance database.

For details about what data can be replicated, refer to 17.2.3, “What’s new in WCS V5.1 LDAP support” on page 476.

17.11 Where to find more information

- *LDAP Implementation Cookbook*, SG24-5110
- *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163
- *e-Commerce Patterns using WebSphere Commerce Suite, Patterns for e-business Series*, SG24-6156
- *User-to-Business Pattern Using WebSphere Personalization Patterns for e-business series*, SG24-6213
- *AIX Version 4.3 Differences Guide*, SG24-2014-02
- IBM SecureWay Directory home page:

  http://www.ibm.com/software/network/directory/
- IBM SecureWay Directory download page:

  http://www.ibm.com/software/network/directory/downloads/
- IBM SecureWay Directory documentation:

  http://www-4.ibm.com/software/network/directory/library/

18

WebSphere Payment Manager

This chapter describes how to install and configure WebSphere Payment Manager V2.2 for use with WebSphere Commerce Suite V5.1. We have included an example Payment Manager runtime configuration, where the Payment Manager server is installed on a separate system from the WCS server and uses a remote DB2 server to host its databases. The example includes procedures for enabling WCS for integration with Payment Manager.

This chapter is organized into the following sections:

- WebSphere Payment Manager overview
- Install Payment Manager
- Configure Payment Manager
- Configure WCS for Payment Manager
- Where to find more information

© Copyright IBM Corp. 2001

18.1 WebSphere Payment Manager overview

This section provides an overview of the Payment Manager hardware and software used within our test environment, sample configurations, and high-level instructions for installation and configuration.

18.1.1 WebSphere Payment Manager hardware

We used the following hardware within our test environment.

AIX platform

We tested with the following hardware on the AIX platform:

- IBM RS/6000 Model 43P, 768 MB RAM, 8 GB hard disk

Windows NT/2000 platform

We tested with the following hardware on the Windows NT/2000 platform:

- IBM xSeries 230, 1 GB RAM, 18 GB hard disk

18.1.2 Payment Manager installation scenarios

There are two main installation scenarios:

- Local Payment Manager server

  In this configuration, Payment Manager is installed on the same server as WCS. The Payment Manager database can be local or remote.

  For additional information on this configuration scenario, refer to the Payment Manager chapter in one of the following product guides:

  - *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*
  - *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*

- Remote Payment Manager server

  In this configuration Payment Manager is installed on a remote server from the WCS server. The Payment Manager database can be local or remote. This configuration scenario is documented in this chapter.

18.1.3 ITSO Payment Manager runtime scenario

Figure 18-1 depicts the Payment Manager runtime configuration used within the ITSO test environment. The instructions to create this type of configuration are documented in subsequent sections.

The DB2 instance db2inst2 includes the WebSphere repository database for Payment Manager, called WASPM, and the Payment Manager database called PAYMAN. The DB2 instance db2inst1 includes the WebSphere repository database for WCS called WAS and the WCS application database called WCS.

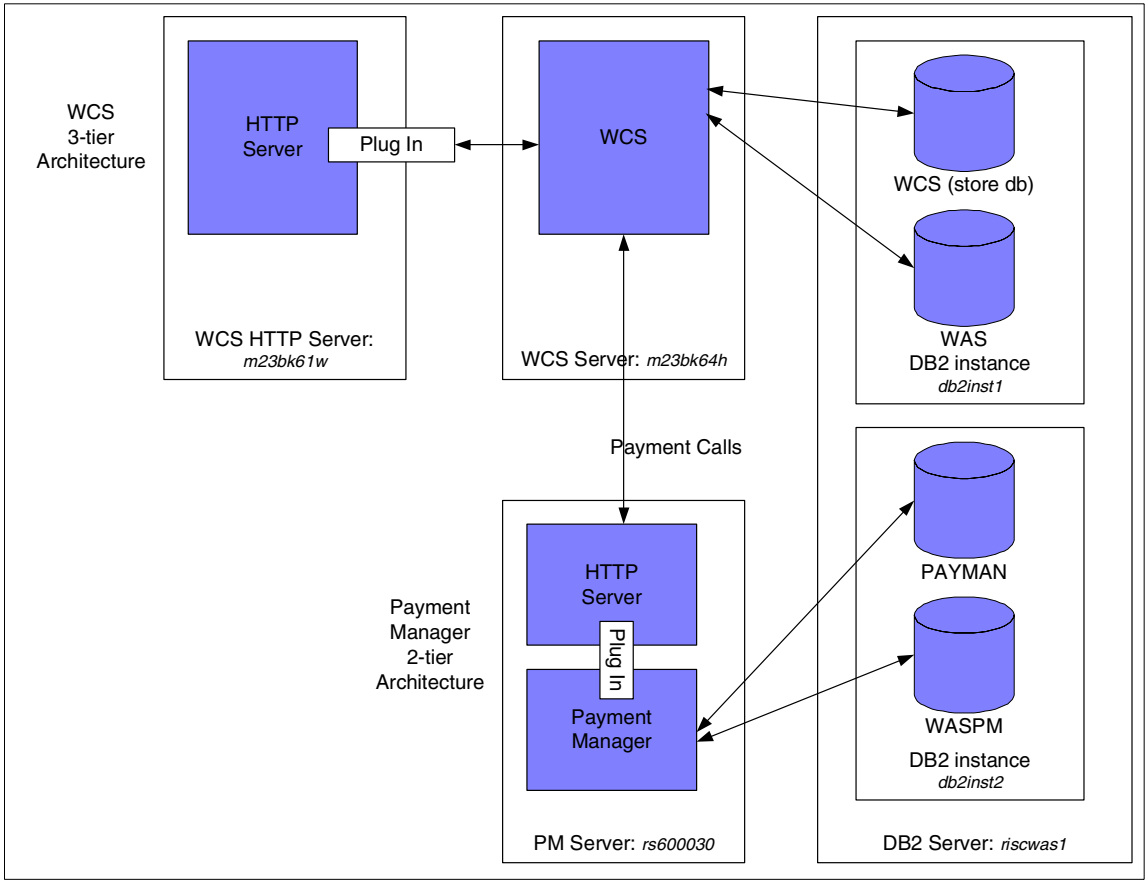


Figure 18-1 ITSO Payment Manager runtime scenario

WCS 3-tier runtime

The ITSO WCS 3-tier runtime is configured as follows:

- WCS HTTP Server

  - Hostname: m23bk61w
- WCS Server
  - Hostname: m23bk64h
  - DB2 instance: db2inst1
- DB2 Server
  - Hostname: riscwas1
  - DB2 instance: db2inst1
  - WCS WAS database: was
  - WCS instance database: wcs

Payment Manager runtime

The ITSO Payment Manager runtime is configured as follows:

- Payment Manager (PM) Server
  - Hostname: rs600030
  - DB2 instance: db2inst2
- DB2 Server
  - Hostname: riscwas1
  - DB2 instance: db2inst2
  - Payment Manager WAS database: waspm
  - Payment Manager database: payman

High-level installation steps

To provide a better understanding of the flow of the installation, we have included the high-level installation steps in Table 18-1.

Table 18-1 High-level install steps - Payment Manager V2.2

Steps	WCS Server (m23bk64h)	PM Server (rs600030)	DB2 Server (riscwas1)
Step 1			Install DB2 server
Step 2		PM prerequisites	

Steps	WCS Server (m23bk64h)	PM Server (rs600030)	DB2 Server (riscwas1)
		PM hardware prerequisites, refer to *Install Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2* Component installation: ► IBM HTTP Server V1.3.12 ► IBM DB2 V7.1 + Fixpack 2a ► IBM WAS V3.5 + Fixpack 2 + E-fixes ► Verify functionality of WAS Pre-installation configuration for Payment Manager: ► Configure DB2 client to access the remote database server (machine C) ► Create payman db ► Start WAS ► Disable WAS security prior to install	
Step 3		Install Payment Manager	
Step 4		Configure Payment Manager ► Configure PM Engine ► Configure PM server ► Configure PM for WCS	
Step 5	Configure WCS for PM		

18.2 Install Payment Manager

This section details the steps to install Payment Manager on a remote system (separate from WCS), which serves a WebSphere Commerce Suite environment.

18.2.1 Install DB2 server

This section describes how the DB2 server must be installed and configured on the remote DB2 server system for the Payment Manager.

1. Ensure that you have increased the file system size for /usr and the /home directory of the DB2 instance.
2. Install DB2 V7.1 Enterprise Server
3. Install DB2 V7.1 Fixpack 2a

Note: Refer to one of the following for detailed instruction on how to install a DB2 server:

- AIX DB2 instructions - 7.3, “Install the DB2 Server” on page 151
- Windows NT/2000 DB2 instructions - 6.3, “Install the DB2 Server” on page 96

4. Created a DB2 instance, called db2inst2, on the DB2 server for Payment Manager. This instance will be used for the Payment Manager WebSphere repository database (WASPM) and the Payment Manager database (PAYMAN).
5. Create the database WASPM for the Payment Manager WebSphere repository. For example:

```
# su - db2inst2
$ db2 create db waspm
$ db2 update db cfg for waspm using applheapsz=512
```

18.2.2 Payment Manager installation prerequisites

This section outlines the prerequisite hardware and software required to be installed for the Payment Manager server system.

Hardware prerequisites

Please refer to the *Install Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2* for complete details on the required software for the installation.

Software prerequisites

The IBM WebSphere Payment Manager V2.2 requires the following software products installed on the Payment Manager server:

- IBM HTTP Server V1.3.12
- IBM DB2 V7.1 Administrative Client
- IBM WebSphere Application Server V3.5.0 + Fixpack2 + E-fixes installed

Note: Refer to Chapter 7, “WCS runtime for AIX: DB2 and IHS” on page 137 for detailed installation instructions.

Install the HTTP Server

For detailed instructions on installing the IBM HTTP Server V1.3.12 for AIX on the Payment Manager server system, refer to 7.2, “Install the IBM HTTP Server” on page 144.

Install the DB2 7.1 Client

Ensure that you have increased the file system size for /usr and the /home directory of the DB2 instance.

For detailed instructions on installing the IBM DB2 UDB V7.1, Enterprise Edition Administrative Client for AIX on the Payment Manager server system, refer to 7.8.3, “Step 3: Install the DB2 client” on page 180.

Install DB2 V7.1 Fixpack 2a

For detailed instructions on installing the DB2 V7.1 Fixpack 2a for AIX on the Payment Manager server system, refer to 7.3.3, “Install the DB2 V7.1 Fixpack 2a” on page 157.

Configure the DB2 V7.1 client

This section outlines the steps involved to create the Payment Manager database. In our environment, the Payment Manager database resides on the Database server machine.

1. Catalog the database server to the Payment Manager server. On the Payment Manager server, type the following:

```
# su - db2inst2
```

```
$ db2 catalog tcpip node riscwas1 remote riscwas1 server 50002
```

2. Verify the attach to the TCP/IP node as follows:

```
$ db2 attach to riscwas1 user db2inst2 using <your_password>
```

3. Catalog Payment Manager WebSphere repository database on the Payment Manager system:

```
$ db2 catalog db waspm at node riscwas1
```

4. Verify the connection to the database from the client to the server as follows:

```
$ db2 connect to waspm user db2inst2 using <your_password>
```

Create the Payment Manager database

On the database machine, create the `payman` database by completing the following steps:

1. Log on to the Database server machine as root and create a terminal window.
2. In this terminal window, create a database using the DB2 instance that the WebSphere Commerce server is using:

```
# su - db2inst2
$ db2 create db payman
```

   This will allow the WCS/WAS server and the PM server to be in the same DB2 instance and therefore be seen by each product.

3. On the Payment Manager server, catalog the payman database on the remote database server:

```
$ db2 catalog db payman at node riscwas1
```

4. Verify the connection to the database from the client to the server as follows:

```
$ db2 connect to payman user db2inst2 using <your_password>
```

This database must be running during the installation process.

Install WebSphere Application Server

For detailed instructions on installing the IBM WebSphere Application Server V3.5, Advanced Edition for AIX on the Payment Manager server system, refer to 7.4, “Install the WebSphere Application Server” on page 157.

WebSphere Application Server configuration

The IBM WebSphere Payment Manager requires the Websphere Application Server administration server to be running.

Important: Document the SSL host entries within default_host by looking at the WebSphere Administrator's Console. During the Payment Manager install, the <hostname>:443 entries are often removed. This requires that the administrator to reenter these entries manually after the Payment Manager installation is completed.

18.2.3 Install Payment Manager

The following steps are required to install the IBM WebSphere Payment Manager product.

1. On the Payment Manager machine, log on as root and create a terminal window.
2. Insert the IBM WebSphere Payment Manager CD-ROM into the CD drive.
3. Mount the CD-ROM as follows:

   ```
   # mount -r -v cdrfs /dev/cd0 /mnt
   ```

4. Start the installation

   ```
   # cd /mnt
   # ./Install
   ```

5. This will launch the Payment Manager Install window. Click **Next** to continue.
6. Review the license agreement, and click **Accept** if you agree to the terms.
7. Accept the default installation path.
8. The installation window will prompt you for the HTTP server publish directory. The default is /usr/HTTPServer/htdocs/en_US, assuming the HTTP server is on the same machine.
9. If the installation cannot determine the Java Technology Edition that WebSphere Application Server is using, then you will be prompted for the location. You should not be asked if you have installed the WebSphere Application Server on the same machine as Payment Manager.
10. Next, select the database that you will be using; in our testing we used DB2.
11. The JDBC driver location is now requested. Accept the appropriate one. Enter the DB2 Instance name, for example `db2inst2`.
12. When the Payment Manager Database Access Information window appears, enter the following and then click **Next**:
    - Database Owner User ID: `<db2_instance_owner>`

      For example, `db2inst2`.

- Database Administrator User ID: <db2_admin_id>

  Depending on your DB2 configuration, this could be the “db2fenc1” user or the DB2 instance owner. We used db2inst2 (db2 instance owner).
- Database Administrator Password: <db2_admin_password>
- Payment Manager Database Name: <pm_database>

  For example, we created a database for Payment Manager called payman in a previous step.

13. The Payment Manager configuration window will now prompt you for the Payment API port number. Accept the default of 8611.
14. The payment installation program will now display the WebSphere Configuration information window. This will be the machine name that is hosting Payment Manager, that is the WebSphere Application Server. For the example in our environment, it was rs600030.
15. The Installation summary window is now displayed. Review the settings, then click **Next** to start the installation.

At this point the IBM WebSphere Payment Manager software is now installed on the PM server machine, and ready to be configured for the store environment.

Note: During the Payment Manager install, the <hostname>:443 entries are often removed. This requires that the admin reenter these entries after the Payment Manager installation is completed.

18.3 Configure Payment Manager

This section describes how to configure Payment Manager for use with WebSphere Commerce Suite.

18.3.1 Post installation configuration

This section details the steps involved when configuring IBM WebSphere Payment Manager V2.2 for 2-tier and 3-tier AIX environments. The configuration for a single-tier AIX environment is the same as a 2-tier environment.

The IBM WebSphere Payment Manager V2.2 software does not detect the presence of WebSphere Commerce Suite, since it exists on another machine. Hence, the configuration of Payment Manager needs to be done manually.

Note: The order of starting the Payment Manager software components is important. The Payment Manager Engine should be started *before* the Payment Servlet.

Configure the Payment Manager Engine

This section details the steps involved in configuring the IBM WebSphere Payment Manager Engine.

1. Log on to the Payment Manager server as root and start a terminal window.
2. Ensure that the DB2 client and WebSphere Application Server software are started on the Payment Manager server.
3. Modify the Payment Manager Engine startup script file.

   The script file name is IBMPayServer.aix. The contents of the original file are shown in Example 18-1, with the items requiring customization highlighted like this: *<<sample>>*. Table 18-2 on page 546 provides sample values used in the ITSO environment for the IBMPayServer.aix file. which can be found in /usr/lpp/PaymentManager.

Example 18-1 Sample IBMPayServer.aix

```
#!/bin/sh
ODBC_SHARED_LIB_PATH=
DB2INSTANCE=<<db2_instance>>
LIBPATH=./lib:<<jdbc_shared_lib_path>>:$ODBC_SHARED_LIB_PATH:$LIBPATH
LD_LIBRARY_PATH=./lib:<<jdbc_shared_lib_path>>:$ODBC_SHARED_LIB_PATH:$LD_LIBRAR
Y_PATH
export DB2INSTANCE
export LIBPATH
export LD_LIBRARY_PATH
CASSETTE_CLASSES=<<CassetteClasses>>
cd <<Payment_Manager_Install_Directory>>
"<<javaExe>>" -noclassgc -classpath
<<jdbc_classes>>:./eTillClasses.zip::$CASSETTE_CLASSES:./eTillxml4j209.jar:"<<j
avaClasses>>":.:$CLASSPATH -"<<jit>>" \
     com.ibm.etill.framework.ETill  \
     DBDriver=<<JDBC_driver_name>> \
     DBjdbcURL=<<JDBC_URL>>        \
     DBOwner=<<Database_owner>>    \
     DBUserID=<<Database_userid>>  \
     DBPassword=$1
```

Table 18-2 Payment Manager server - Engine customization

Fields to be customized	Value used in this redbook
<<db2instance>>	db2inst2
<<jdbc_shared_lib_path>>	/usr/lpp/db2_07_01/lib
<<CassetteClasses>>	eTillCustomOfflineCassetteClasses.zip eTillOfflineCardCassetteClasses.zip The filenames are separated by colons
<<Payment_Manager_Install_Directory>>	/usr/lpp/PaymentManager
<<javaExe>>	/usr/jdk_base/bin/java This is the JDK included with AIX V1.1.8
<<jit>	remove entry
<<jdbc_classes>>	/usr/lpp/db2_07_01/java12/db2java.zip
<<java_Classes>>	/usr/jdk_base/lib/classes.zip
<<JDBC_driver_name>>	COM.ibm.db2.jdbc.app.DB2Driver
<<JDBC_URL>>	jdbc:db2:payman Where payman is our Payment Manager Database.
<<Database_owner>>	db2inst2
<<Database_userid>>	db2inst2
DBPassword	$1 By default this can be supplied as a parameter (see Step 4 below)

4. The IBMPayServer.aix script file accepts one parameter, which is the DB2 instance owner password. An alternative to entering the password on the command line is to store the password in a file called .payment. The .payment file is created as a read-only file and is hidden as much as the operating system allows.

   Once this startup file has been modified with the appropriate password, the Payment Manager Engine can be started. To start the Payment Manager Engine with the .payment file, complete the following steps:

5. Make sure the .payment file contains the correct password for the db2 instance owner.

6. Change the directory to the Payment Manager install directory and start the Payment Manager Engine:

   ```
   # cd /usr/lpp/PaymentManager
   ```

```
# nohup ./IBMPayServer &
```

Tip: Using the **nohup** command allows the Payment Manager Engine to remain active when you log off of the Payment Manager server. Any output is directed to the nohup.out file in the Payment Manager directory.

7. To verify that the server is running, run the following command and look for the message `Websphere Payment Manager has started successfully:`

```
# cd /usr/lpp/PaymentManager
# tail -f nohup.out
```

Configure Payment Manager server

This section details how to configure the IBM WebSphere Payment Manager server component to work with the WebSphere Commerce Suite environment. Follow the steps below to complete the configuration for the server component of IBM WebSphere Payment Manager:

1. Log on to the Payment Manager machine as root and start a terminal window.
2. Ensure that the WebSphere Admin Server (startupServer.sh) and the WebSphere Administrative Console (adminclient.sh) are started.
3. Using the WebSphere Administrative Console, stop the WebSphere Payment Manager application server.
4. Modify the /usr/lpp/PaymentManager/PaymentServlet.properties file.

   This file needs to be modified so that the Payment Manager is aware of the WCS realm.

   a. Change the RealmClass property to the full class name of the WCSRealm class:

   ```
   RealmClass=com.ibm.commerce.payment.realm.WCSRealm
   ```

   b. Add the following properties:

   ```
   WCSHostName=<WCS_HOST_NAME>
   WCSWebPath=/webapp/wcs/stores/servlet
   ```

Note: In a 3-tier environment, the <WCS_HOST_NAME> value is the name of the HTTP server. This is the result of separating the WAS/WCS and HTTP server function. In a single-tier and 2-tier environment, where the WAS/WCS and HTTP server functionality is not split, the value of <WCS_HOST_NAME> is the WAS/WCS hostname.

In our example the WCS_HOST_NAME is M23BK61W.

   c. Once these changes have been made, save the file.

5. Copy the wcspmrealm.jar file from the WCS machine onto the Payment Manager machine into the installation directory (by default: /usr/lpp/PaymentManager). Rename this file to: PMRealm.jar.
6. Add wcsadmin user to the PSRealm as follows:

```
# cd /usr/lpp/PaymentManager
# ./PSDefaultRealm PSRealm add wcsadmin wcsadmin
```

7. As documented earlier in this chapter, the <hostname>:443 entries may have been removed from the default_host entry in the WebSphere Application Server Administration client. Restore the entries as documented in the pre-installation steps.
8. Restart the Websphere Payment Manager application server from the WebSphere Administrative Console.

At this stage, we have installed and configured the Payment Manager application server on a separate machine. As a result of these steps, the WebSphere Commerce Suite Administrator ID, wcsadmin, is automatically assigned the Payment Manager administrator role.

Note: Do not delete or rename the logon user called wcsadmin and do not change the pre-assigned Payment Manager role of wcsadmin.

18.3.2 Configure Payment Manager for a WCS Store

This section outlines the steps involved in configuring a store to use IBM WebSphere Payment Manager V2.2. In this example, we configure a store for offline payments with a VISA card.

1. Enter the following URL to start the Payment Manager.

   `http://<Your_payment_server>/PaymentManager`

2. Enter the username of wcsadmin, and wcsadmin's password.
3. This will bring up the Payment Manager Web interface. Click **Merchant Settings**.
4. Click **Add Merchant**.
5. When the Merchant Settings window appears, enter the following and then click **Create Merchant**.
   - Merchant name: `WCS <store_name> merchant`

     For example, WCS WebFashion merchant.
   - Merchant number: `<wcs_store_id>`

Tip: The Store ID can be found in the STOREENT table of the store database. This can found by using the following commands:

```
db2 connect to <store_dabasename>
db2 select * from STOREENT
```

- Authorized cassettes: select **OfflineCard**

  The authorized cassette list will vary depending on the number of cassettes you have installed.

6. When the message `Merchant created successfully` appears, click **Merchant Settings** from the left-hand navigation bar.
7. Click the **OfflineCard** icon, which is green in color.
8. Click **Accounts**.
9. Click **Add an Account**.
10. Enter the Merchant account information, as follows and then click **Create Account**.
    - Account name: `Offline account 1`
    - Account number: `100011`

      We recommend that you use the <store_id><number> that is unique.
    - Financial Institution name: `ITSO Bank`
    - Currency: select **US Dollar**
    - Batch close time: `0`
11. Click the account you created (for example, Offline account 1).
12. Click **Brands**.
13. Click **Add a Brand**.
14. Enter a credit card brand name (for example, we entered `VISA`) and then click **Create Brand**.

    These steps can be repeated to create other brands such as Master Card, American Express, and Diners Club.

18.4 Configure WCS for Payment Manager

This section describes how to configure and verify the WebSphere Commerce Suite instance for integration with Payment Manager.

Configure the WCS instance

To configure the WCS instance using the Configuration Manager for use with Payment Manager, complete the following steps:

1. Ensure that the IBM WCS Configuration Manager Server Windows service is started.
2. Start the WCS Configuration Manager on the WebSphere Commerce server.
3. After logging on, click **<wcs_instance>** -> **Instance Properties** -> **Payment Manager** tab.
4. When the Payment Manager tab appears, enter the following:
   - Hostname: <payment_manager_hostname>.

     For a multi-tier configuration, this is the fully qualified hostname of the Payment Manager server Web server. For example, rs600030.itso.ral.ibm.com.
   - Accept the defaults for the remaining values and then click **Apply**.
5. Click **OK**, and then close the Configuration Manager.

Note: We recommend that you stop the IBM WCS Configuration Manager Server Windows service when done for security reasons and to reduce memory usage.

6. Stop and start the WebSphere Commerce Server - <wcs_instance> application server from the WebSphere Administrative Console for the changes to take effect.

Verify the WCS and Payment Manager configuration

To verify that WCS is configured properly with Payment Manager, complete the following steps:

1. Enter the following URL to start the WCS Administration Console:

   ```
   http://<wcs_webserver_hostname>/adminconsole
   ```
2. Log on to the Administration Console as wcsadmin.
3. From the Administration Console, select **Payment Manager** -> **Merchant Settings**.

   You should see the merchant settings that were created on the Payment Manager server.
4. Enter the WCS store URL in a Web browser, for example:

   ```
   http://m23bk61w/webapp/wcs/stores/servlet/StoreCatalogDisplay?storeId=10001
   &langId=-1&catalogId=10001
   ```

5. Log on as a registered user and add an order to the shopping cart. When checking out, you should have the credit card brand options (for example, VISA, Master Card, AMEX) that you entered on the Payment Manager server.

> **Tip:** For testing purposes, select VISA and enter the following credit card:
>
> 4111 1111 1111 1111 (The number 4 followed 15 1s).
>
> Alternatively, use the VISA number: 1111222233334444.

6. Log on to the Payment Manager remote administration console as follows to verify that the payment record has been received.

   ```
   http://<payment_manager_hostname>/PaymentManager
   ```

7. Log on to Payment Manager as wcsadmin.
8. From the left-hand navigation bar, select **Order Search**.
9. When the Order Search window appears, enter the desired criteria and click **Search**. You should see your order.

Congratulations, your WCS environment is now enabled to use Payment Manager.

18.5 Where to find more information

- WebSphere Commerce Suite V5.1 online help
- *Administrator's Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2*, found on the product CD.
- *Install Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2* found on the product CD.
- IBM WebSphere Payment Manager V2.2 home page:

  ```
  http://www.ibm.com/software/webservers/commerce/paymentmanager/
  ```

- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*

19

WCS messaging using MQSeries and e-mail

Typically, in an enterprise environment, WebSphere Commerce Suite will need to exchange details such as order activity and shipment status with existing applications (legacy). A reliable message transport between WebSphere Commerce Suite and back-end systems is implemented using the IBM MQseries asynchronous messaging product.

This chapter provides an overview of the Commerce Suite messaging functions and describes how MQSeries and e-mail can be configured to provide message transports for WebSphere Commerce Suite.

This chapter is organized into the following sections:

- WebSphere Commerce Suite messaging
- MQSeries installation
- Create the WCS MQSeries queues
- JMS installation and configuration
- WCS MQ adapter configuration
- WCS MQ adapter verification
- e-mail messaging enablement

© Copyright IBM Corp. 2001

19.1 WebSphere Commerce Suite messaging

WebSphere Commerce Suite V5.1 provides support for integration with back-end systems and notification of events via messaging subsystems for inbound and outbound messaging.

For example, an e-mail note can be sent to a customer by the outbound messaging system acknowledging receipt of an order or notifying an error. Similarly, an XML formatted message can be generated and sent via MQSeries (MQ) to notify a back-end fulfillment system that an order has been placed so that an inventory update and shipment processes can be performed.

Similarly, the fulfillment system could at various stages of the fulfilment process generate and send messages to Commerce Suite via MQ and the inbound messaging system to update the shipping status of the order in Commerce Suite

WebSphere Commerce Suite provides a set of predefined messages and associated programming and also allows additional messages to be created (with some programming effort).

19.1.1 Message transports

Inbound messages are received into WebSphere Commerce Suite via the MQ Adapter.

Outbound messages can be sent as e-mail via an SMTP server or as MQSeries messages via an MQSeries queue manager. There is also an option to have the messages written to a file on the local file system. This file could then be transferred to another system via FTP or processed by a local application.

WebSphere Commerce Suite V5.1 implements back-end messaging by using the Java Message Service (JMS) API, which provides a generic interface to an asynchronous messaging system such as MQ.

This chapter will focus on how to configure a back-end communication link using MQSeries.

19.1.2 Messaging and queueing with IBM MQSeries

IBM MQSeries is a middleware product that provides communications between applications using an asynchronous queuing interface. Applications may be on the same physical machine or on separate systems connected via a network link.

Queue managers on the connected systems provide functions for storing messages on queues where they can placed and retrieved by applications. Messages can be placed on local queues or remote queues (on remote systems). There is no dependency on the remote system being available at the time of message transmission, since the message will be stored until such time as the destination system becomes available.

The MQ API (MQI) is an application programming interface that allows application programs to put and retrieve messages to/from queues via a set of MQ commands. The programmer can get and put messages to a remote queue without having to be concerned with the nature of the underlying communication link.

MQ is available for all major software platforms and the MQI API supports most common programming languages.

The Java Messaging Service (JMS) API is the industry-standard Java interface for Message Oriented Middleware (MOM). WCS uses JMS for connectivity for MQSeries.

19.1.3 Inbound messages

An inbound message is a message that Commerce Suite receives from an external application via the MQSeries adapter. Each inbound message activates a command in Commerce Suite that performs a particular function. If there is an error processing an inbound message, the MQSeries adapter places the failed message in the error queue.

The MQSeries adapter supports inbound messages that accomplish the following five functions:

- Create a customer registration using the Create_WCS_Customer XML message and the **`UserRegistrationAdd`** controller command.
- Update a customer registration using the Update_WCS_Customer XML message and the **`UserRegistrationUpdate`** controller command.
- Update the status of an order using the Update_WCS_OrderStatus XML message and the **`OrderStatus`** controller command.
- Update the inventory for a product using the Update_NC_productInventory XML message and the **`ProductInventoryUpdate`** controller command.
- Update the price of a product using the Update_WCS_ProductPrice XML message and the **`ProductOfferPriceUpdate`** or **`ProductListPriceUpdate`** controller command.

For more information, refer to the WCS online help.

19.1.4 Outbound messages

An outbound message is a Commerce Suite-generated message that can be sent to an external system.

A predefined message Report_NC_PurchaseOrder XML allows you to communicate to back-end systems whenever an order is generated. The XML message is encoded in Unicode UTF-8 format. You can also use the legacy Commerce Suite Order Create message instead, which performs a similar function.

The outbound messages with order information can be used to send order information from the WebSphere Commerce Suite server to external systems, where further order fulfillment processes take place.

To enable the outbound message, first you need to choose which one you want to use, either the Report_NC_PurchaseOrder XML message or the legacy Commerce Suite Order Create message. Note that the two cannot be enabled at the same time.

19.1.5 High level steps to enable MQSeries messaging

WebSphere Commerce Suite V5.1 requires several components to be installed and configured for MQSeries messaging. There are two modes that MQSeries can be implemented within a WCS environment.

- MQSeries client-server mode
- MQSeries bindings mode (client-server, server)

MQSeries client-server mode

MQSeries client can be installed and configured on the WCS server with connectivity to a remote MQSeries server, as seen in Figure 19-1. This configuration requires minimal resources. The disadvantage to this mode is that the Java Native Interface (JNI) cannot be explored for communication. If the server is down, communication may not be reliable.

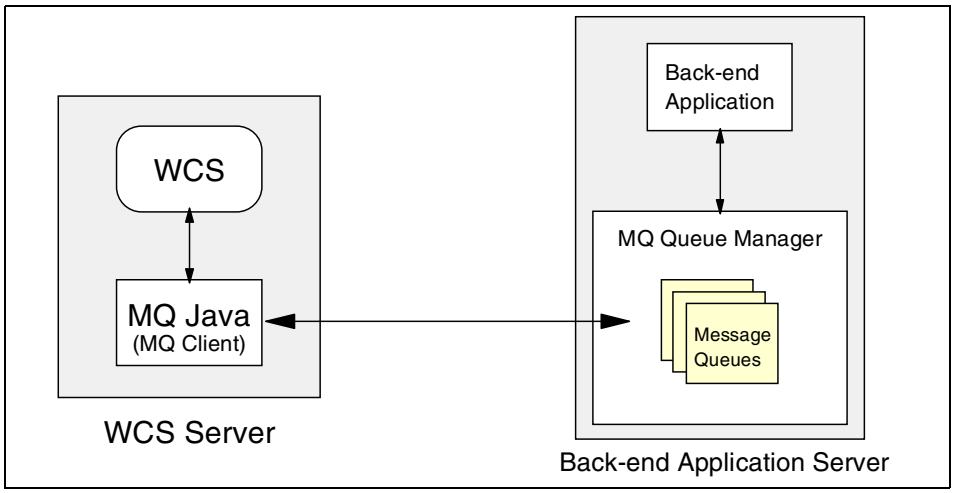


Figure 19-1 MQSeries client-server mode

MQSeries bindings mode

MQSeries server and queues can be installed and configured on the WCS server. This type of configuration is called bindings mode, as seen in Figure 19-2.

This configuration mode requires more machine resources (for example, disk space for message queues), but will remove dependency on the synchronous connection to the remote queue manager. This will provide a more robust system since, if the connection to the remote system fails, MQ will hold outbound messages on a queue until the remote system is available.

An additional advantage to thIs configuration mode is that the MQSeries Java client can connect directly to MQSeries on the local machine using the Java Native Interface (JNI) and the MQSeries base Java API (MQI, bindings mode).

.

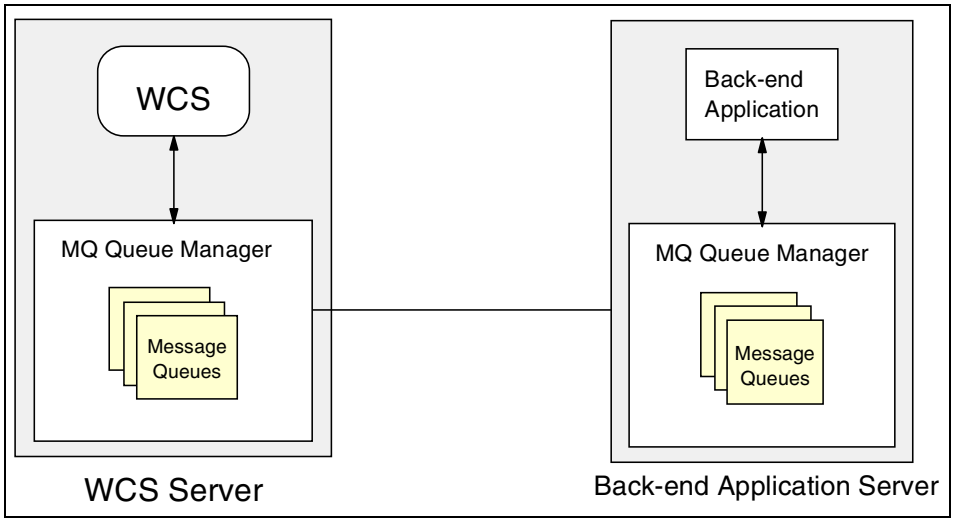


Figure 19-2 MQSeries bindings mode

The WebSphere Commerce Suite MQSeries adapter, or simply the MQSeries adapter, is a component of WebSphere Commerce Suite that enables integration with back-end systems.

To enable the MQSeries adapter you must have MQSeries and MQSeries classes for Java Message Service (JMS) installed on your system and you must have the MQSeries adapter and the messaging system configured on your WCS system.

The following list includes the high-level steps for installation, which are detailed in subsequent sections of this chapter:

- Step 1: MQSeries installation
- Step 2: Create the WCS MQSeries queues
- Step 3: JMS installation and configuration
- Step 4: WCS MQ adapter configuration
- Step 5: WCS MQ adapter verification

19.2 MQSeries installation

Officially, WebSphere Commerce Suite V5.1 supports MQSeries V5.1. In our test runtime environment, we used MQSeries V5.21 for Windows NT/2000 and MQSeries V5.2 for AIX. The installation procedure in this section documents the MQSeries bindings mode configuration.

Note: There are some side effects and issues to take note of if you decide to use MQSeries V5.2 or choose to use MQSeries V5.1. First, MQSeries V5.1 will be phased out by the end of 2001 and the MA88_xx.zip is no longer available for V5.1.

If you choose to use MQSeries V5.2, this version is not officially supported at this time. Also, there are some manual configurations within VisualAge for Java that will need to be made. At the heart of this issue is the WCS5101.dat, which includes MQSeries V5.1 support.

MQSeries V5.2 requires a minimum of approximately 40 MB of disk space for product code and data. In addition, allow a minimum of 20 MB for working space. MQSeries also requires that Microsoft MMC and Microsoft ADSI be installed first. This software is provided on the MQSeries Server CD-ROM.

The installation procedure documented in this section is for the Windows NT/2000 platform with the MQSeries server installed on the WCS server. The concepts can be applied to a remote MQSeries server and installations on other operating system platforms, such as AIX.

For more information, refer to the MQSeries Quick Beginnings manual at:

`http://www.ibm.com/software/ts/mqseries/library/manuals/index.htm`

MQSeries installation

To install MQSeries V5.21 on a Windows NT Server, complete the following steps:

1. Install the prerequisite software, MMC and ADSI, provided on the MQSeries Server CD-ROM.

   Using Windows NT Explorer, open the Prereqs folder on the MQSeries Server CD-ROM. Within the Prereqs folder you will find the MMC and ADSI folders. Start the installation executable for each for the desired language to install the components. Accept the installation defaults.

2. From the Windows Explorer, run `setup.exe` from the MQSeries V5.21 Server CD-ROM. Refer to the *MQSeries V5.2, Quick Beginnings Guide for Windows NT and Windows 2000* for details on the installation procedure.

Note: When using a WCS messaging, we recommend you do not install MQSeries in the default directory. Instead, install MQSeries on the same path where WCS is installed and use a directory name without spaces.

For example, we installed MQSeries in the c:\ibm\mqseries directory.

Verify the MQSeries installation

Verify the installation using the MQSeries Postcard application to create two postcards on the same machine and send messages between them. This verifies that MQSeries messaging is working correctly. The MQSeries server installation is now complete.

Detailed information can be found in the *MQSeries Quick Beginnings Guide*.

19.3 Create the WCS MQSeries queues

You must define a queue manager and inbound, outbound, and error queues for use with the WebSphere Commerce Suite messaging system. You can do this either by using the MQSeries command line tool or by using the MQSeries Explorer tool. Using the MQSeries Explorer tool, the steps to take are:

1. Select **Start** -> **Programs** -> **MQSeries** -> **MQSeries Explorer**.
2. To create a queue manager, right-click the **Queue Managers** folder and select **New** ->**Queue Manager**. Enter the queue manager name, for example, `wcs_mq_qmgr`. We made this our default queue manager. Accept the default for the remaining options.

   Alternatively, you can use this if you created a default queue manager when installing MQSeries.
3. Once the queue manager is created, you need to create the queues that will be used by WebSphere Commerce Suite.
   a. From the MQSeries Explorer, select and expand **Queue Managers** -> **wcs_mq_qmgr** -> **Queues.**
   b. Right-click the **Queues** folder and select **New** -> **Local Queue** and complete the Queue Name field. WCS requires that you configure five queues. We used the following queue names:
      - wcs_inbound
      - wcs_outbound
      - wcs_error
      - wcs_inbound_par
      - wcs_inbound_ser

4. You must set the coded character set identifier of your MQSeries queue manager to 1208 (UTF8) by typing the following MQSeries commands from the command line within the <MQ_install_path>\bin directory (for example, c:\ibm\mqseries\bin):

```
strmqm YourQueueManagerName
runmqsc YourQueueManagerName
alter qmgr ccsid(1208)
end
```

Figure 19-3 displays a view of the MQSeries Explorer after the queues have been created for WCS.

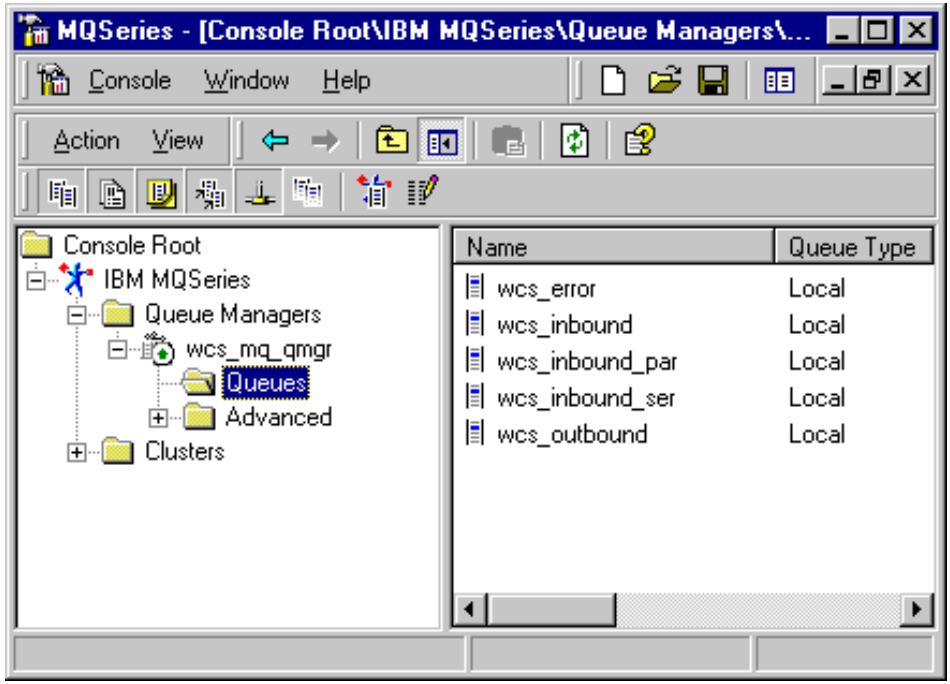


Figure 19-3 MQSeries Explorer with WCS queues defined

19.4 JMS installation and configuration

The MQSeries V5.2 SupportPac MA88 contains the IBM MQSeries classes for Java and the IBM MQSeries classes for Java Message Service. This support pack must be installed to provide a mapping between JMS generic objects and the queue manager and queues used by MQSeries.

To install this SupportPac, complete the following steps:

1. Download the SupportPac code from:

   http://www.ibm.com/software/ts/mqseries/txppacs/ma88.html

2. The code is in compressed format (zipped). Decompress (unzip) the file and run **setup.exe**.

   Follow the setup instructions and install the product extensions in the <MQ_install_path>\java directory.

   Where <MQ_install_path> is the path that MQSeries is installed.

Note: To change the default directory, select the **Custom** option and change the install directory to your MQSeries installation directory.

3. Update the Windows NT system classpath and path variables following the instructions in Chapters 2 and 4 in the *MQSeries Using Java* manual.

   a. Required JMS classpath entries:

      c:\ibm\mqseries\java\lib\com.ibm.mq.jar

      c:\ibm\mqseries\java\lib\com.ibm.mqjms.jar

      c:\ibm\mqseries\java\lib\jms.jar

      c:\ibm\mqseries\java\lib\jndi.jar

      c:\ibm\mqseries\java\lib\jta.jar

      c:\ibm\mqseries\java\lib\ldap.jar

      c:\ibm\mqseries\java\lib\providerutil.jar

      Where c:\ibm\mqseries in the MQSeries install path.

   b. Optional JMS classpath entries:

      c:\ibm\mqseries\java\lib\com.ibm.mq.iiop.jar

      c:\ibm\mqseries\java\lib\com.ibm.mqbind.jar

      c:\ibm\mqseries\java\lib\fscontext.jar

   c. Required WebSphere Application Server JMS classpath entry:

      c:\ibm\was\lib\ujc.jar

   a. Optional WAS JMS classpath entry:

      c:\ibm\was\lib\ejs.jar

   b. Required JMS Windows system path entry:

      c:\ibm\mqseries\java\lib

   c. Required WebSphere Application Server Windows system path entrys:

      c:\ibm\was\jdk\jre\bin

Note: We recommend that you insert the entries at the beginning of the classpath and path.

4. Add a new environment variable named MQ_JAVA_INSTALL_PATH and set it to:

   MQ_JAVA_INSTALL_PATH=c:\ibm\mqseries\java

   Where c:\ibm\mqseries\java is the MQSeries JMS install path.

5. Ensure that the IBM JRE is included in your path. To test that the JRE is configured properly from a command, enter the following:

   ```
   java -fullversion
   ```

   This should return a message with the IBM build and level.

6. We recommend that you verify your installation using the IVTRun program following the instructions in Chapter 4 in the MQSeries Using Java manual.

   Example:

   ```
   IVTRun -nojndi -client -m wcs_mq_qmgr -host m23vnx58
   ```

   The output should state that the put and get of the message were successful.

   Syntax:

   ```
   IVTRun -nojndi -client -m <qmgr> -host <hostname>
   ```

7. Map the queue manager and queues that you have created in MQSeries to the WCS namespace by completing the following steps:

   a. Make sure that the WebSphere Application Server is running and that the correct classpath and environment variable have been created.

   b. Change to the <MQ_install_path>\java\bin directory (for example, c:\ibm\mqseries\java\bin).

   c. Open the JMSAdmin.config file in a text editor.

   d. Ensure that the following three variables have been set to the following in order to connect to the WAS PNS:

      ```
      INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory
      PROVIDER_URL=iiop://localhost:900
      SECURITY_AUTHENTICATION=none
      ```

Note: The desired entries must be uncommented, and the following default values must be commented out.

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
PROVIDER_URL=ldap://polaris/o-ibm,c=us
```

e. Run the JMSAdmin program by typing the following line at a command prompt:

```
JMSAdmin -cfg JMSAdmin.config -t -v
```

Wait for the administration command line interface to load and the Initctx> prompt to appear.

f. Register your queue connection factory to your queue manager in the WebSphere Application Server namespace by typing the following command at the prompt:

```
define qcf(JMSQueueConnectionFactory) qmanager(YourQueueManagerName)
```

g. Then set the coded character set identifier to 1208 (UTF8) by typing the following command at the prompt:

```
alter qcf(JMSQueueConnectionFactory) ccsid(1208)
```

Where `JMSQueueConnectionFactory` is the name of the MQQueueConnectionFactory JMS object (MQSeries Keyword: QCF), which is the MQSeries implementation of the JMS QueueConnectionFactory interface as defined in the inbound transports in the Configuration Manager, and `YourQueueManagerName` is the name of your MQSeries queue manager.

8. You need to define the following JMS queues:
   - JMSSerialInboundQueue for your serial inbound queue.
   - JMSParallelInboundQueue for your parallel inbound queue.
   - JMSInboundQueue for your inbound queue.
   - JMSOutboundQueue or your outbound queue.
   - JMSErrorQueue or your error queue.

   For example, for your serial inbound queue, type the following:

```
define
q(JMSSerialInboundQueue)qmanager(YourQueueManagerName)queue(YourSerialInbou
ndQueueName)
```

   Where `YourSerialInboundQueueName` is the name of your MQSeries queue created for your serial inbound queue (for example, `wcs_inbound`).

9. For your outbound queue and error queue you need to set the target client to indicate that JMS is dealing with a native MQSeries application by typing the following lines:

```
alter q(JMSOutboundQueue) targclient(MQ)
alter q(JMSErrorQueue) targclient(MQ)
```

Note: This step applies only to the MQSeries client/server configuration.

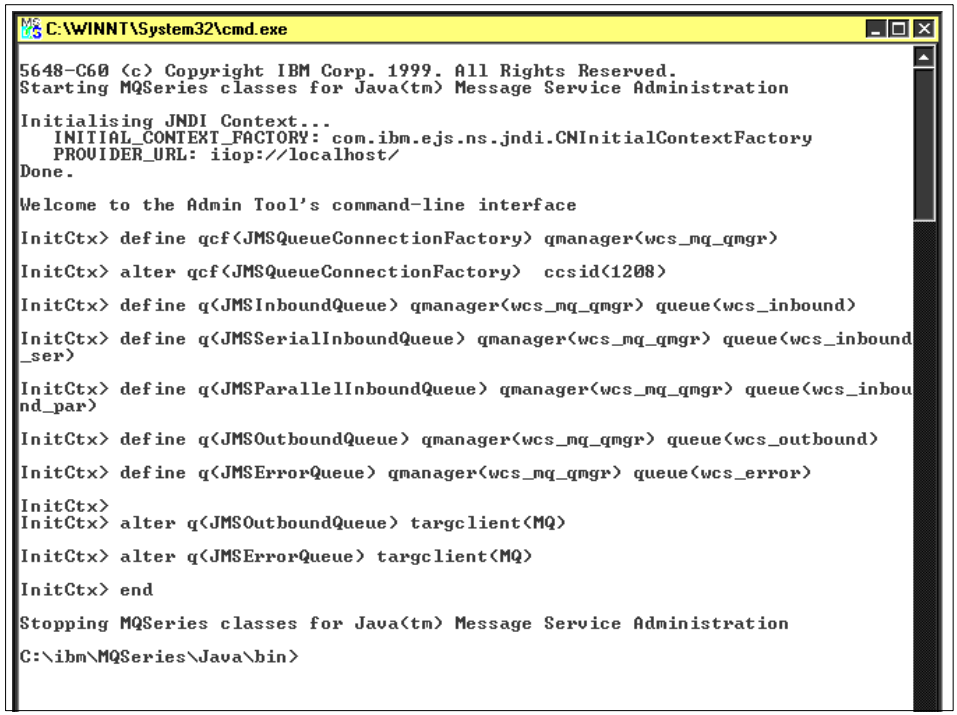
10. If you are using an MQSeries client/server setup and your MQSeries client is on the same machine as your WebSphere Commerce Server, you need to configure JMS so that it uses the MQSeries client transport. Enter the following JMS commands:

```
alter qcf(JMSQueueConnectionFactory) transport(CLIENT)
alter qcf(JMSQueueConnectionFactory) hostname(YourMQServerHostName)
```

    Where `YourMQServerHostName` is the name of your MQSeries server.

11. Exit the JMSAdmin tool by typing `end`.

Figure 19-1 displays sample values for the JMS objects configuration used for our environment.



Example 19-1 Sample commands used to create JMS queues

19.5 WCS MQ adapter configuration

To use WCS MQSeries messaging, you must enable the messages system transport adapter, update the WebSphere Application Server classpath, and add some settings in your WCS instance and store.

19.5.1 Enable the messaging system transport adapter

To enable the messaging system transport adapter, or MQSeries adapter, do the following:

1. Open Configuration Manager. From the Start menu select **Programs -> WebSphere Commerce Suite -> Configuration**.
2. Select your instance, then open the **Components** folder.
3. Select **TransportAdapter**.
4. Ensure that the checkbox next to Enable Component is activated and click **Apply** (see Figure 19-4).

Note: These changes will take effect the next time you start the WebSphere Commerce Server - <wcs_instance> application server from the Administrator's Console.

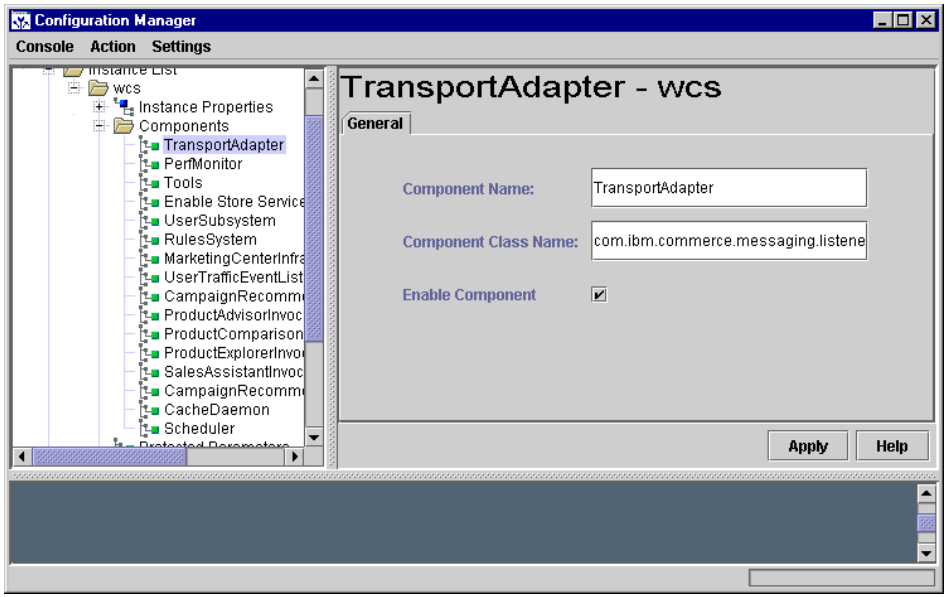


Figure 19-4 WCS Configuration Manager: enable transport component

19.5.2 Update the WAS classpath variable

To update the WebSphere Application Server classpath variable for an instance, do the following:

1. Open the WebSphere Administrative Console.

2. Select the host on which you are running your Commerce Suite instance.
3. Select the WebSphere Commerce Suite Application Server instance_name node, where instance_name is the name of your Commerce Suite instance.
4. In the Command line arguments field, append the following text, all on one line:

   ```
   -classpath
   <MQ_install_path>/java/lib/;<MQ_install_path>/java/lib/com.ibm.mq.jar;<MQ_i
   nstall_path>/java/lib/com.ibm.mqjms.jar;<MQ_install_path>/java/lib/jms.jar;
   <MQ_install_path>/java/lib/providerutil.jar;<MQ_install_path>/java/lib/fsco
   ntext.jar;
   ```

   Where `<MQ_install_path>` is the path MQSeries is installed.

Note: Make sure there are no spaces in the path. Otherwise the WCS instance will not start. This is not a problem if you followed previous instructions to not have spaces in the installation path.

5. Click **Apply** to apply the changes.

Note: This must be done for every WCS instance that uses MQSeries.

19.5.3 Configure WCS transports and messaging

This section provides configuration instructions for transports and messaging.

WCS transports configuration

To configure the WCS transports and messages, do the following:

1. Start the WCS Administration Console by clicking **Start** -> **Programs** -> **IBM WebSphere Commerce Suite** -> **Administration Console**.
2. Choose the **Site** radio button.
3. Select **Messaging** -> **Transports**.
4. Check the row for MQSeries and click the **Change Status** option (see Figure 19-5).

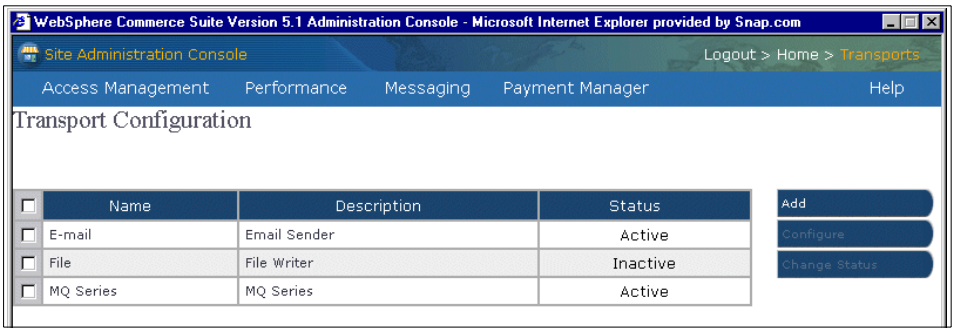


Figure 19-5 Administration Console: transport configuration

5. Select the MQSeries row and click the **Configure** option. Ensure the parameters as seen in Figure 19-6.

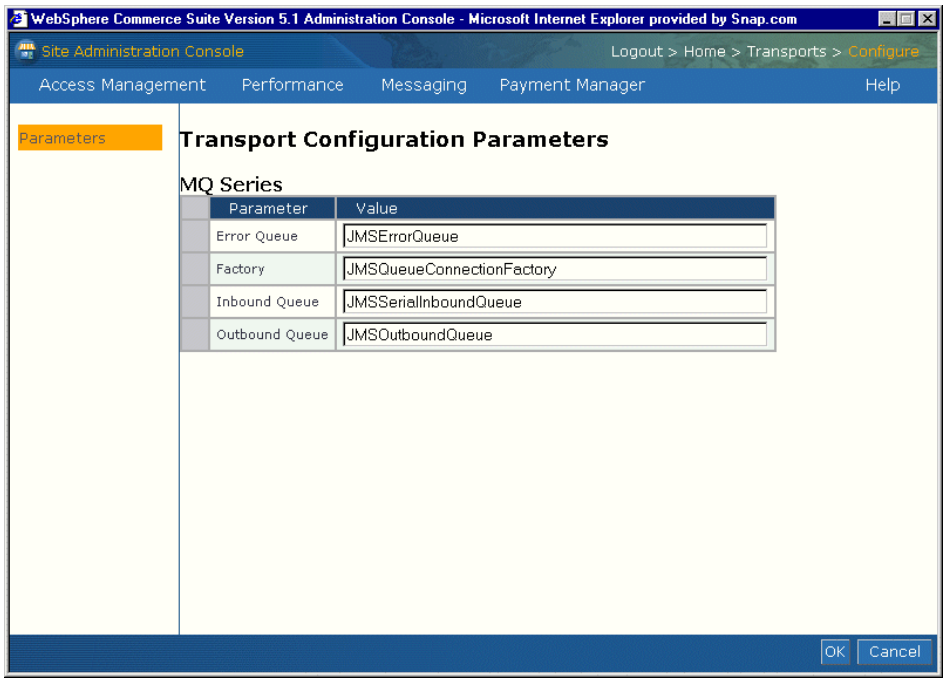


Figure 19-6 Administration Console: transport configuration parameters

WCS messaging configuration

To configure messaging, complete the following steps:

1. Select **Messaging** from the menu and choose the **Message Types** option.

2. Select the message type **Outbound message for WebSphere Commerce Suite order create**.
3. Click **Change** as shown in Figure 19-7.

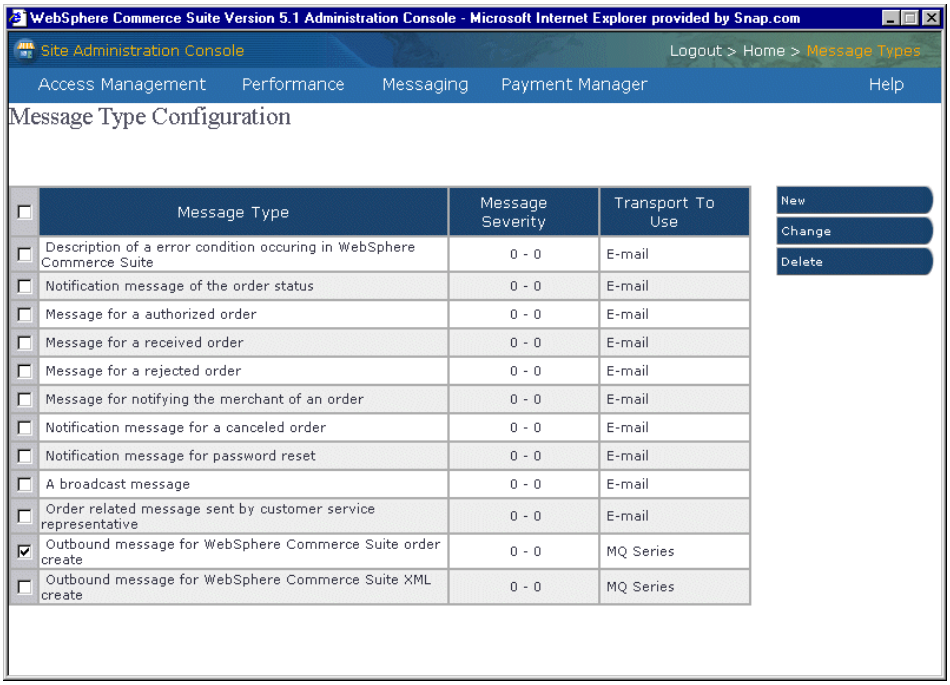


Figure 19-7 Administration Console: messaging configuration

4. The message transport assignment window is now displayed. In this window you can assign the transport and configure your outbound message (see Figure 19-8).

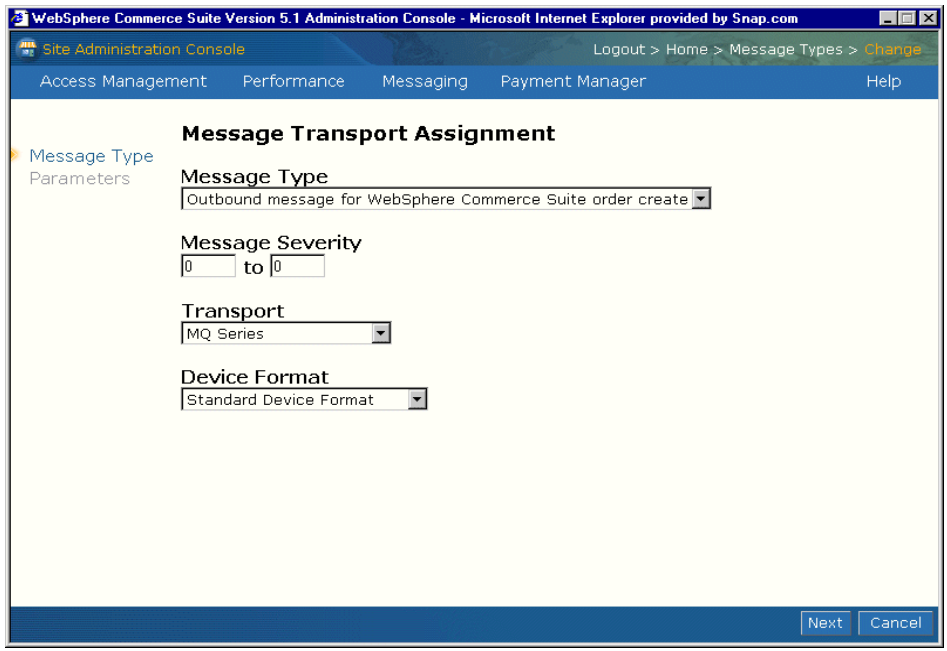


Figure 19-8 Administration Console: message transport assignment

5. When the Message Transports Assignment window appears, ensure that the parameters are as seen in Figure 19-9.

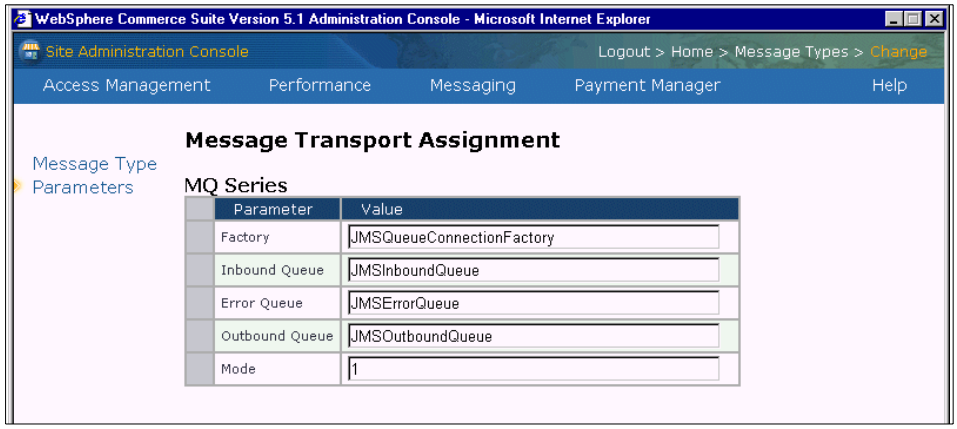


Figure 19-9 Administration Console: message transport assignment

19.5.4 Additional settings in WCS

Enabling outbound messages is necessary to do some changes on your instance database. Use either the DB2 command center or the DB2 command window to make the following changes:

6. Connect to <database_of_instance>.
7. Obtain the store entity ID. In order to do that, you could execute the following SQL:

   ```
   select * from storeent
   ```

   Select the storeent_ID value that corresponds to your store.

8. Register the command **SendXMLOrder** to use it in the store:

   ```
   insert into CMDREG(STOREENT_ID, INTERFACENAME, CLASSNAME, DESCRIPTION,
   TARGET) values (<storeent_ID>,
   'com.ibm.commerce.order.commands.OrderMessagingCmd',
   'com.ibm.commerce.messaging.commands.SendXMLOrderCmdImpl', 'Order Create
   XML', 'Local')
   ```

9. Update the VIEWREG table for OrderOKView to use an HTTP redirect:

   ```
   update VIEWREG set VIEWNAME='OrderOKResultView' where
   STOREENT_ID=<storeent_ID> and VIEWNAME='OrderOKView'
   insert into VIEWREG (STOREENT_ID, DEVICEFMT_ID, VIEWNAME, INTERFACENAME,
   CLASSNAME, PROPERTIES, HTTPS, INTERNAL) values (<storeent_ID>, -1,
   'OrderOKView', 'com.ibm.commerce.command.RedirectViewCommand',
   'com.ibm.commerce.command.HttpRedirectViewCommandImpl',
   'redirecturl=OrderOKResultView', 0, 0)
   ```

10. To tell the system to use the OrderCreateXML.jsp at the Mall level:

    ```
    update viewreg set properties='docname=OrderCreateXML.jsp&storeDir=no'
    where viewname like '%OrderCreateXML%' and storeent_id=0
    ```

11. If you are using the InFashion store and you do not install the Payment Manager, you must enable the store to generate orders without using the Payment Manager.

    To do that, execute:

    ```
    update cmdreg set classname =
    'com.ibm.commerce.payment.commands.DoPaymentCmdImpl' where interfacename =
    'com.ibm.commerce.payment.commands.DoPaymentCmd'
    ```

    In addition, you need to change the OrderDisplayPending.jsp code in the CREDIT_CARD_TYPE section. You could have something like this:

    ```
    <td align="left" valign="middle">
    <font class="required">*</font>
    <font class="strongtext">
    <%=infashiontext.getString("CREDIT_CARD_TYPE")%>
    </font>
    ```

```
</td>
<td align="left" valign="middle">
<select name="cardBrand" >
<option value ="visa">VISA</option>
<option value ="mast">MASTER</option>
<option value ="amex">AMEX</option>
</select></td>
```

19.6 WCS MQ adapter verification

To verify the configuration, complete the following steps:

1. Ensure that the MQ queue manager is running.

   Tip: Set the MQSeries queue manager within MQ services to start automatically.

2. Ensure that the transport adapter is checked and applied within the WCS Configuration Manager.
3. Restart the WCS instance.
4. Verify from the Administration Console that the MQSeries transport type is active.
5. Create an order using the checkout process of your store (see Figure 19-10).

   Tip: For testing credit card orders, use Visa credit card number 0 or 4111111111111111 (15 1s).

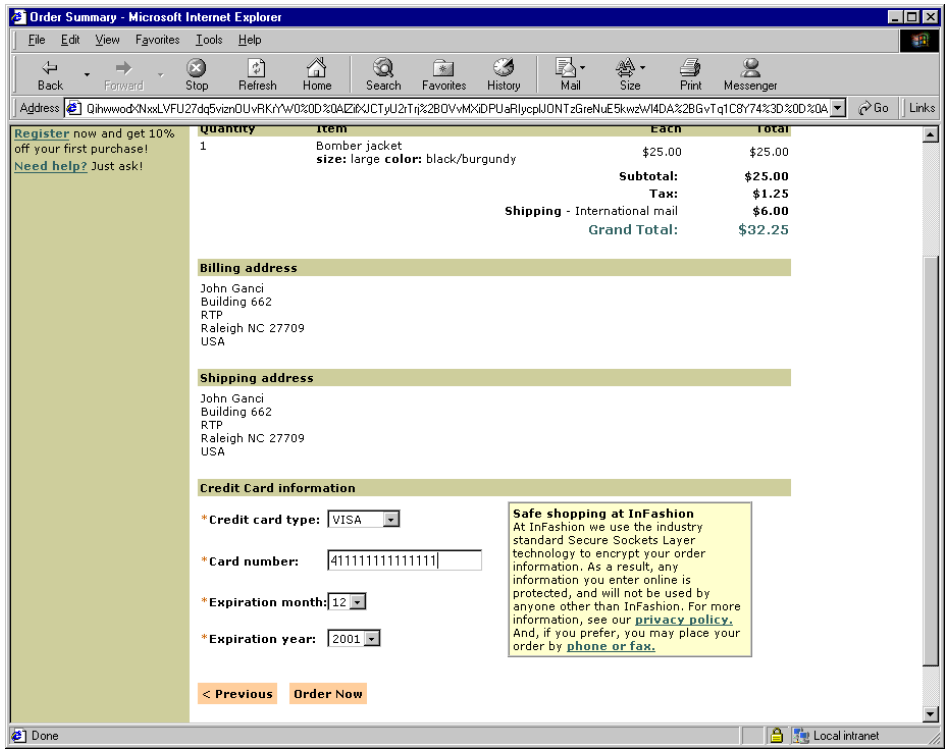


Figure 19-10 Order page

6. Check that the message is in the instance database, using the following SQL:

   ```
   select * from MSGSTORE
   ```

   The MSGSTORE table is used for temporary storage and holds the messages stored as a result of using a transacted send.

   When you execute the SQL, you must have “1” row and in the retries field, you could have the value “3”. Wait a moment and re-execute the SQL. When your select retrieves “0” rows, go to the next step.

7. Browse the outbound message on MQSeries.

   a. Select **Start** -> **Programs** -> **IBM MQSeries** -> **MQSeries Explorer**.

   b. Select **Queue Managers** -> **yourQueueManager** -> **Queues**.

   c. Right-click the queue **wcs_outbound** and select **Browse Messages**.

      You should see the newly created outbound order message like Figure 19-11.

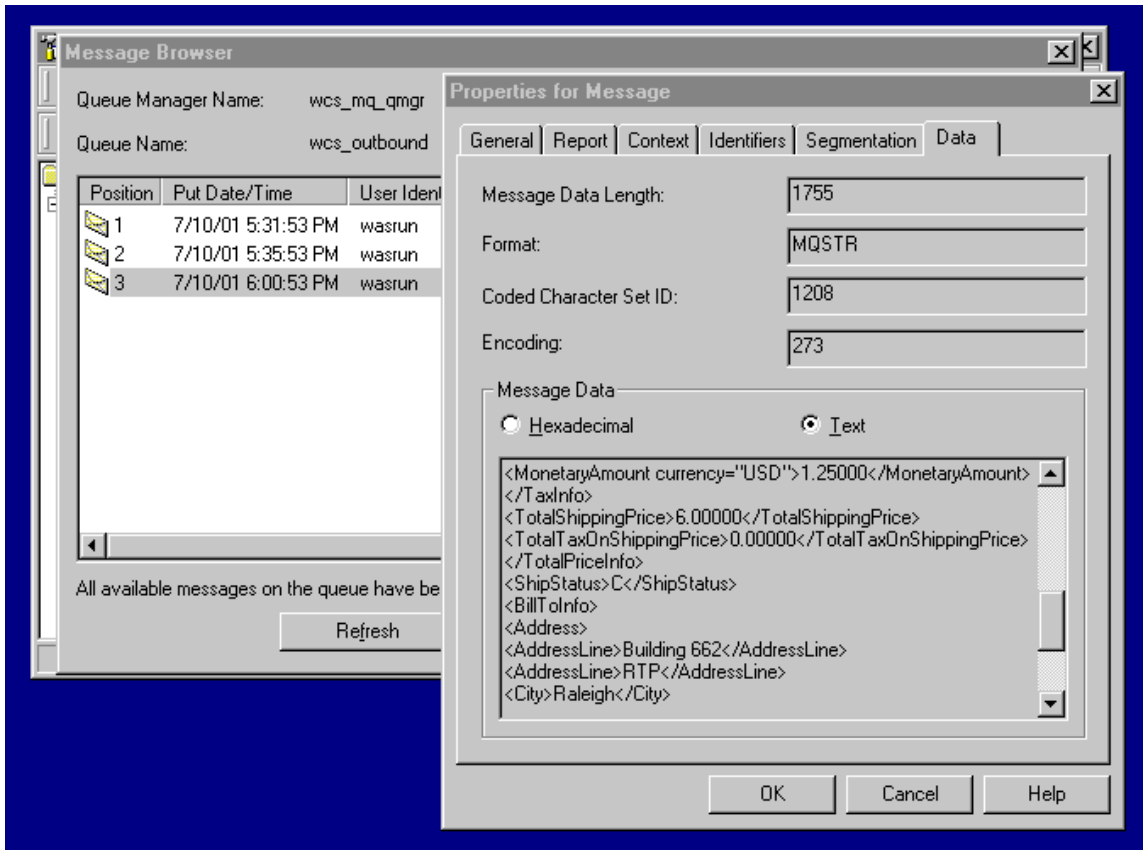


Figure 19-11 WCS MQ outbound order message

8. If you have any problems with the MQ adapter, enable the WCS log system configuration, within the Configuration Manager, as shown in Figure 19-12.

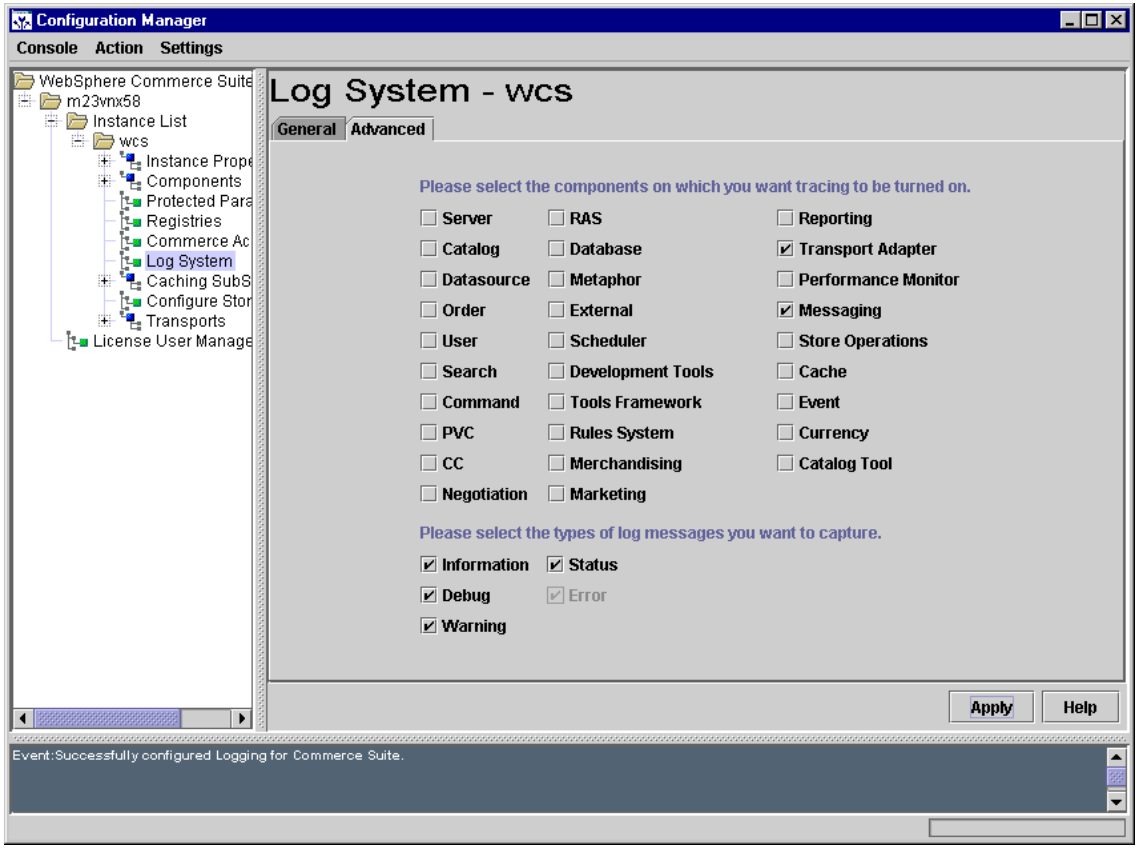


Figure 19-12 WCS Configuration Manager: Log System settings

9. After restarting the WCS instance, check the following log files:
   - <wcs_instance\logs\wcs.log

     This log file will have information about the MQ adapter initialization.
   - <wcs_instance\logs\ecmsg_xx.log

     This log file will have information about the messaging runtime.
   - <MQ_install_path>\java\log

     If logging is enabled for MQSeries JMS, log files will be generated in this directory.

19.7 e-mail messaging enablement

This section describes how to enable WCS e-mail messaging and is organized into the following phases:

- Prerequisites for e-mail notification
- WCS e-mail transport configuration
- Configuring messages for e-mail notification
- Verify e-mail messaging configuration

19.7.1 Prerequisites for e-mail notification

The WCS e-mail messaging enablement requires that you have an SMTP mail server and POP3 mail server. The SMTP mail server should be enabled to receive inbound e-mail.

To test the configuration, we used the following:

- Lotus Notes
- IMail Server

  An evaluation copy of this product can be downloaded at:

  http://www.ipswitch.com

Note: We recommend that you test the mail server for both SMTP and POP3 services prior to configuring WCS for e-mail messaging.

19.7.2 WCS e-mail transport configuration

To configure e-mail for WCS, complete the following steps:

1. Start the WCS Administration Console by entering the following URL:

   `http://<hostname>/adminconsole`
2. Log on with the site administrator user ID and password (for example, wcsadmin).
3. Select **Site**, and click **OK**.
4. When the Site Administration Console window appears, select **Messaging** -> **Transports**.
5. In the Transport Configuration window, check **E-mail** and then click **Configure**.
6. In the Transport Configuration Parameters window, enter the following and then click **OK**:

– Host: <smtp_hostname>

  For example, in our test configuration we set up the SMTP server and POP3 server as localhost.

– Protocol: smtp

7. When the Transport Configuration window appears, ensure that the e-mail status is active.

19.7.3 Configuring messages for e-mail notification

To configure messages for e-mail notification, complete the following steps:.

1. Start the WCS Administration Console by entering the following URL:

   `http://<hostname>/adminconsole`

2. Log on with the site administrator user ID and password (for example, wcsadmin).
3. Select **Site**, and click **OK**.
4. When the Site Administration Console window appears, select **Messaging** -> **Message Types**.
5. When the Message Type Configuration window appears (Figure 19-13), insert the text from step 6. on page 578.

   We will demonstrate how to configure the notification message for password reset. The same procedure can be applied to other messages whose transport type is e-mail.

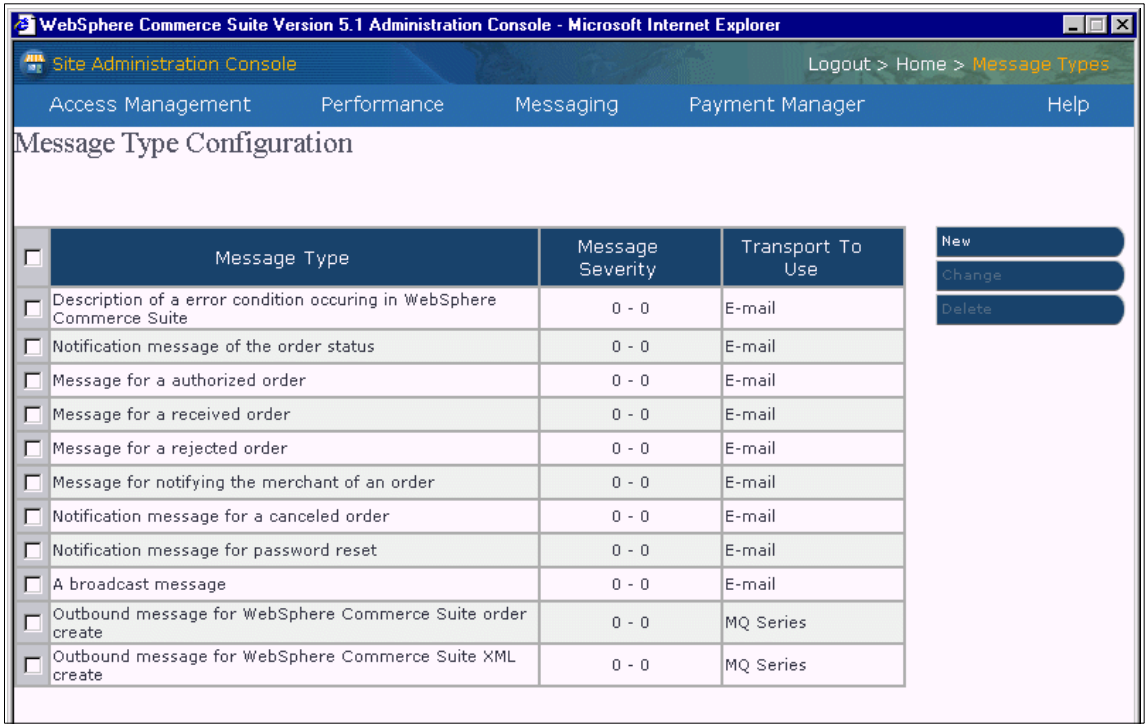


Figure 19-13 WCS outbound messages

6. Check **Notification message for password reset**, and then click **Change**.
7. When the Message Transport Assignment window appears, ensure the following parameters are set and then click **Next**:
   - Message Type: `Notification message for password reset`
   - Message Severity: `0 to 0`
   - Transport: `E-mail`
   - Device Format: `Standard Device Format`
8. When the Message Transport Assignment for E-mail window appears, enter the following and click **Finish:**
   - Host: `<smtp_hostname>`

     For example, in our test environment we used localhost.
   - Protocol: `smtp`
   - Recipient: `recep@ibm.com`

     This value is a placeholder. The actual recipient will be the user e-mail address.

- Sender: `wcsadmin@<smtp_hostname>`
- Subject: `Password reset`

9. Close the Administration Console.
10. Stop and start the WebSphere Commerce Server - <wcs_instance>.

19.7.4 Verify e-mail messaging configuration

To verify that the e-mail messaging configuration is working properly, complete the following steps:

1. To test the password reset, enter your store URL.
2. Register a user with an e-mail address that is configured for your SMTP mail server.

Note: If you are using WebFashion as your base store, WebFashion requires that discounts be enabled. To enable discounts, refer to the Store creation and customization chapter in the *WebSphere Commerce Suite V5.1 Handbook*, SG24-6167.

3. Log off or close the Web browser window.
4. Enter the store URL, and click the **Register** button.
5. Click the **Forgot your password?** link.
6. When the Forgot your password page appears, enter your e-mail address that you previously registered and click **Send me my password**.
7. You should now see a message saying that your password has been sent.
8. From your e-mail client, verify that your new password has been received.
9. Test your new password by logging into your store.

Part 6

Appendixes

© Copyright IBM Corp. 2001

A

WebSphere Application Server tips

This appendix includes WebSphere Application Server tips on the following topics:

- Creating a new WAS database
- WAS log files

© Copyright IBM Corp. 2001

Creating a new WAS database

If your WCS instances become corrupt or if you just want a clean WAS database, you might consider deleting your WAS database and creating a new one. This may not be a good option if you have made many configurations changes to the WAS database via the WebSphere Adminsistrator's Console.

Complete the following steps:

1. Stop the IBM WS AdminServer from Windows Services.
2. Start a DB2 command window.
3. Type the following command to delete the database:

   ```
   > db2 drop db was
   > db2 create db was
   > db2 update db cfg for was using applheapsz 256
   ```

   Where `was` is the WebSphere Application Server repository database.
4. Change to the directory <was_install_path>\bin,. for example, c:\ibm\was\bin.
5. Modify admin.config and change the lint to `install.initial.config=true`.
6. Start the IBM WS AdminServer from Windows Services.
7. Verfiy a successful start by reviewing the <was_install_path>\logs\tracefile.
8. Start the WebSphere Administrator's Console.
9. Start the Default Server.
10. Enter URL: `http://<hostname>/servlet/snoop` in a Web browser.
11. Re-add any configuration changes made to WAS via the WebSphere Administrator's Console.
12. Stop the Default Server to conserve memory.
13. Recreate your WCS instance.

WAS log files

This section contains information on WAS log files.

Tracefile

The tracefile can be found in the <was_install_path>\logs directory. This file is useful to find out if WebSphere Application Server started successfully.

Look for the following message if successfully started:

```
AdminServer   A WebSphere Administration server open for e-business
```

B

Solaris tips

This appendix provides basic procedures, tasks, and commands necessary to install, configure, and manage the Solaris operating system in preparation for a WebSphere Commerce Suite V5.1 installation.

The appendix is organized into the following sections:

- Solaris 7 installation
- Common Solaris tasks and commands
- Where to find information about Solaris

© Copyright IBM Corp. 2001

Solaris 7 installation

The WebSphere Application Server V3.5, Advanced Edition requires Solaris 7 (2.7) for the Sun SPARC platform. The most current WebSphere hardware, software, and API for V3.5 requirements can be found at:

http://www-4.ibm.com/software/webservers/appserv/doc/v35/prereq.html

The Solaris 7 installation is organized into the following sections:

- Planning for Solaris 7 installation
- Solaris 7 base installation

  This phase installs the required base operating system files and configures the system with basic network and locale information.
- Solaris 7 interactive installation

  This phase provides the installation of selected software options and file system layouts.
- Solaris 7 post install configuration

  This phase provides the necessary steps to configure Solaris for common server usage.
- Solaris 7 patches required for WCS V5.1

Planning for Solaris 7 installation

During the Solaris 7 installation you will be prompted for information specific to your environment. We have included the necessary information needed to complete the installation in preparation for installing WCS V5.1 Pro for Solaris.

Network information

The following network information is needed during the installation:

- Hostname
- Primary network
- IP address
- Domain name
- Subnet netmask

File system planning

During the interactive phase of the Solaris 7 operating system installation, you will need to customize the file systems and allocate the necessary space for the applications to be installed in subsequent steps.

For example, if you building a 2-tier WCS runtime for Solaris with Oracle and iPlanet Web Server, your file system requirements will be as follows.

Machine A - WCS Server (iPlanet, WAS, WCS, Oracle client)

Table B-1 provides general guidelines for the space required on the WCS Server by application.

Table B-1 Diskspace needed - WCS server by application

Product	File system installed to	Disk space needed
iPlanet Web Server V4.15	/usr	130 MB
WebSphere Application Server V3.5	/opt	250 MB
WebSphere Commerce Suite V5.1	/opt	250 MB
Oracle8i database client	/opt	350 MB

Machine B - Database Server (Oracle server)

Table B-2 provides general guidelines for the space required on the WCS Server by application.

Table B-2 Disk space needed - Oracle database server

Product	File system installed to	Disk space needed
Oracle8i database server	/opt	1020 MB
Oralce8i database * WAS * WCS * PM	/opt	800 MB per database. In our example, we create a database called WCS, WAS, and PM.

Solaris 7 base installation

To install Solaris 7, complete the following steps:

1. Insert the Solaris 7 Software CD in the CD-ROM drive.
2. Power on the system and do one of the following:
   - If this is an existing system do the following:
     i. Enter the maintenance mode

ii. Log in as root. In a Solaris terminal window (command prompt) type `halt`.

iii. When OK is displayed, type `boot cdrom` to boot from the Solaris installation CD.

– If the system is new out of the box, turn the system on and it will boot from the CD.

3. When the Select Language and Locale window appears, select the language (for example **English**) and select the locale (for example **USA (ASCII)**), and then click **Continue**.
4. When the Solaris Installation Program window appears, click **Continue**.
5. When the Identify This System window appears, click **Continue**.
6. When the Host Name window appears, type `<your_hostname>` in the Host Name field, and then click **Continue**.
7. When the Network Connectivity window appears, select the **Yes** radio button next to Networked, and then click **Continue**.
8. When the IP Address window appears, type the IP address in the IP Address field, and then click **Continue**.
9. When the Confirm Information window appears, review the entries and then click **Continue**.
10. When the Name Service window appears, select the **DNS** radio button next to Name Service, and then click **Continue**.
11. When the Domain Name window appears, type the domain name in the Domain Name field, and then click **Continue**.
12. When the DNS Server Addresses window appears, enter the DNS IP address in the Server's IP address field, and then click **Continue**.
13. When the DNS Search List window appears, enter the `search domain` in the Search Domain fields, and then click **Continue**.
14. When the Confirm Information window appears, review the entries and then click **Continue**.
15. When the Subnets window appears, select the **Yes** radio button next to System part of a subnet, and then click **Continue**.
16. When the Netmask window appears, type the netmask in the Netmask field, and then click **Continue**.
17. When the Time Zone window appears, select the **Geographic region** radio button next to Specify timezone by, and then click **Set...**.

18. When the Geographic Region window appears, select the **Region** (for example **United States)** and your time zone from the Time Zones list, and then click **Continue**.
19. When the Date and Time window appears, enter the appropriate date and time, and then click **Continue**.
20. When the Confirm Information window appears, review the entries and then click **Continue**.

You have now completed the base installation for Solaris 7.

The following section enters into the Solaris interative installation phase of installing the Solaris 7 operating system.

Solaris 7 interactive installation

Complete the following steps for the Solaris interactive installation:

1. When the Solaris Interactive Installation window appears, click **Continue**. In our example, we selected **Initial** install to simulate a clean system load.
2. When the Solaris Interactive Installation preview window appears, click **Continue** to proceed with the installation.
3. When the Allocation Client Services window appears, click **Continue**.
4. When the Select Languages window appears, click **Continue**. If there are any additional languages that are needed, select the language, click **Add** and continue.
5. When the Select Software window appears, select **Developer System Support**, and then click **Continue**.

Note: Depending on the needs of your runtime environment, you may want to select a different Software Group.

6. When the Select Disks window appears, accept the default, and then click **Continue**.
7. When the Preserve Data window appears, click **Continue**.
8. When the Automatically Layout File Systems window appears, click **Auto Layout**.
9. Next select the following file systems to create by default, and then click **Continue**.
   - / (root)
   - /opt (file system used by Oracle, WAS, WCS)

– /usr

– swap

10. When the File Systems and Disk Layout window appears, click **Customize**.

11. When the Customize Disks window appears, reallocate the disk space to provide a minimum as seen in Table B-3, and then click **OK**.

    For example, the hard disk in our system is 17 GB, 1 GB memory.

Table B-3 Space allocation for Solaris file systems

File system	Description	Usage	Example
/	root	512 MB	512 MB
swap	swap file	Recommend 1-2 times the size of physical memory (min 512 MB)	2048 MB
overlap	na	na	17269 MB
/opt	program files file system used by WAS, WCS	* WAS * WCS * Oracle8i	5120
/usr	* iPlanet Web Server	* iPlanet Web Server 130 MB	2048
/export/home	User home directory	Home directory of users	7534
		total disk space	17269

12. When the File System and Disk Layout window appears again, click **Continue**.

13. When the Mount Remote File Systems window appears, click **Continue**.

14. When the Profile window appears, review the selections and then click **Begin Installation**.

15. When the After Solaris software is installed message windown appears, click **Auto Reboot**.

    The Installing Solaris Software - Progress window appears. This process takes approximately one hour with the options selected.

16. After the installation is complete and the system reboots, you will be prompted to define the password for root.

17. When the System Identification window appears, the message `Do you want this automatic power-saving shutdown?` will be displayed. Type `n` for no, and then press Enter.
18. When the message `Do you want the system to ask about this again?` appears, type `n` and then press Enter.
19. When the Login window appears, type the user name `root`, and press Enter. Next type the root password.
20. When the Welcome to Solaris window appears, select **Common Desktop Environment (CDE)**, and then click **OK**.

The Solaris 7 interactive install is now complete.

Solaris 7 post install configuration

This section provides instructions for the basic configuration steps after installation.

Default router

During the installation, most of the TCP/IP network configuration options are entered, with the exception of one. Without configuring the default router, the system will not be able to communicate beyond the subnet gateway.

To configure the default router, complete the following steps:

1. Start a terminal console session.
2. Create a default router file in the /etc directory:
   a. Change to the /etc directory.

      ```
      # cd /etc
      ```

   b. Create a file called “defaultrouter”:

      ```
      # vi defaultrouter
      ```

   c. Add the gateway IP address for the subnet (for example 9.24.105.1).
   d. Save the file.
3. Restart the system:

   ```
   # ./usr/sbin/shutdown -i6 -g0 -y
   ```

Solaris 7 patches required for WCS V5.1

Complete the following steps to download and install the Solaris 7 patches required for a WCS V5.1 Pro for Solaris:

1. Start a Solaris Console window and enter the following commands:

   ```
   # cd /opt
   # mkdir SolarisPatch
   ```

2. We used the Hot Java Web browser included with Solaris 7 to download the Solaris 7 patch. In order to use the browser, we needed to configure a SOCKS server. Alternatively, you could download, install, and configure a Netscape Web browser (see 8.2.1, “Install Netscape Communicator” on page 214).

3. Enter the following URL in a Web browser to download the Solaris patches:

   `http://www.sun.com/bigadmin/patches/index.html`

4. Click **Solaris 7** for the recommended patches to download to /opt/SolarisPatch.

Note: Make sure you have enough disk space available on the target file system for the download file.

5. From the Solaris Console window command prompt, type the following commands to unzip the 7_Recommended.zip patch file:

   ```
   # cd /opt/SolarisPatch
   # unzip 7_Recommended.zip
   ```

   This creates a directory called 7_Recommended.

6. To install the patches, type the following commands:

   ```
   # cd 7_Recommended
   # ./install_cluster
   ```

7. When prompted with the message `Are you ready to continue with install (y/n):` type `y` for Yes, and then press Enter.

Note: During the installation some of the patches may fail to install.

8. Ensure the following Solaris 7 patches are installed at the indicated level or higher:
   - 106950-07
   - 106980-13
   - 107078-18
   - 107081-25
   - 107636-05
   - 107607-01

Verify the following WCS V5.1 required patches are installed by typing the following command:

```
# showrev -p | grep <patch_name>
```

Where <patch_name> must be included without the -<level> listed above.

For example:

```
# showrev -p | grep 106950
```

9. Shut down the system by typing the following command:

```
# ./usr/sbin/shutdown -i6 -g0 -y
```

Common Solaris tasks and commands

This section provides examples for some common commands and tasks performed using Solaris 7.

Shutdown

1. Start Solaris Console window.
2. Type the following command to shut down and reboot:

```
# ./usr/sbin/shutdown -i6 -g0 -y
```

Tail

Tail is used to view a log file that is being written to.

```
# tail -f <log_file_name>
```

mkdir (create directory)

To create directories, use **mkdir** as follows:

```
# mkdir test
```

The following command will create the t1 within test without having to change to that directory:

```
# mkdir -p /test/t1
```

ps - process list

The ps utility is used to display the process list of running processes with the ID or pid.

Example usage:

```
# ps -ef | awk '{print $1 "\t" $2 "\t" $8}' | more
# ps -ef | grep java
```

id

This command displays the UID and GID of the user logged in.

```
# id
```

Where to find information about Solaris

- For general info, patches, documentation, downloads, and discussions, refer to:

  http://www.sun.com/bigadmin/

- Solaris 7 documentation found at:

  http://docs.sun.com/ab2/products_C/INDEX/@ProductViewer/5423/*;td=1?Ab2Lang=C&Ab2Enc=iso-8859-1#CurrentPos

- Collection of installation information on Solaris 7 found at:

  http://docs.sun.com/ab2/coll.214.4/@Ab2CollView?Ab2Lang=C&Ab2Enc=iso-8859-1

- Collection of systems administrator information on Solaris 7 found at:

  http://docs.sun.com/ab2/coll.47.8/@Ab2CollView?Ab2Lang=C&Ab2Enc=iso-8859-1

- Solaris patches:

  http://www.sun.com/bigadmin/patches/index.html

- Mark G. Sobell, *A Practical Guide to Solaris*, Addison Wesley, 1999, ISBN 020189548X

- Janice Winsor, *Solaris 7 Reference*, Sun Microsystems Press, A Prentice Hall Title, 1999, ISBN 0130200484

C

Oracle tips

This appendix includes basic Oracle8i tasks and commands as well as useful sources of Oracle8i information.

This appendix is organized into the following sections:

- Oracle8i commands and tasks
- Where to find more information

© Copyright IBM Corp. 2001

Oracle8i commands and tasks

This section includes common commands and tasks used within an Oracle8i environment.

Oracle8i server

This section provides instructions for starting and stopping an Oracle8i server.

Starting the Oracle8i server

To start the Oracle8i server, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

```
# su - oracle
```

2. On the Oracle8i server, type the following command:

```
$ svrmgrl
$ connect internal
$ startup
$ quit
```

Stopping the Oracle8i server

To stop the Oracle8i server, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

```
# su - oracle
```

2. On the Oracle8i server, type the following command:

```
$ svrmgrl
$ connect internal
$ shutdown normal
$ quit
```

Oracle8i listener

Starting the Oracle8i listener

To start the Oracle8i listener on the database server, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

```
# su - oracle
```

2. On the Oracle8i server, type the following command:

```
$ lsnrctl start <listener_name>
```

For example:

```
$ lsnrctl start LISTENER
```

Note: All listeners will be started if the listener name is omitted.

Stopping the Oracle8i listener

To stop the Oracle8i listener on the database server, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

   ```
   # su - oracle
   ```

2. On the Oracle8i server, type the following command:

   ```
   $ lsnrctl stop <listener_name>
   ```

   For example:

   ```
   $ lsnrctl stop LISTENER
   ```

Note: All listeners will be stopped if the listener name is omitted.

Check status of the Oracle8i listener

To check the status of the Oracle8i listener on the database server, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

   ```
   # su - oracle
   ```

2. On the Oracle8i server, type the following command:

   ```
   $ lsnrctl status <listener_name>
   ```

   For example:

   ```
   $ lsnrctl status LISTENER
   ```

Note: All listeners' status will be displayed if the listener name is omitted.

Launching the Net8 Configuration Assistant

Net8 enables services and their applications to reside on different computers and communicate as peer applications. The main function of Net8 is to establish network sessions between a Oracle8i client and server. The Net8 Configuration Assistant is a utility used to create or modify Net8 configuration settings.

To start the Net8 Configuration Assistant, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

   ```
   # su - oracle
   ```

2. On the Oracle8i server, type the following command to start Net8:

```
$ netasst
```

Launching the Database Configuration Assistant

To launch the Database Configuration Assistant, complete the following steps:

1. Log on to the Oracle8i database server as the oracle user:

```
# su - oracle
```

2. On the Oracle8i server, type the following command to start the Database Configuration Assistant:

```
$ dbassist
```

Where to find more information

- General information can be found on Oracle Web site at:

  http://www.oracle.com

- Oracle8i Enterprise Edition documentation can be found at:

  http://otn.oracle.com/docs/products/oracle8i/doc_index.htm

  - *Oracle8i Concepts, Release 2 (8.16)*, A76965-01
  - *Oracle8i Installation Guide, Release 2 (8.16) for Sun SPARC*, A77181-01
  - *Oracle8i Administrator's Guide, Release 2 (8.16) for Sun SPARC*, A77184-01
  - *Net8 Administrator's Guide, Oracle8i Release 8.16*, A76933-01

D

Network Dispatcher

This appendix describes how the IBM WebSphere Edge Server Network Dispatcher can be used for scalability within your runtime environment. By designing a runtime environment with multiple Web servers in a cluster (horizontal scaling) the workload can be distributed and additional servers added to the cluster as necessary to meet your scalability needs.

The appendix is organized into the following sections:

- WebSphere Edge Server overview
- Network Dispatcher overview
- Network Dispatcher installation (AIX)
- Network Dispatcher administration
- Session affinity in Network Dispatcher
- Where to find more information

© Copyright IBM Corp. 2001

WebSphere Edge Server overview

The IBM WebSphere Edge Server V1.0 consists of two products:

- IBM Network Dispatcher V3.0
- IBM Web Traffic Express V1.0

Network Dispatcher

The Network Dispatcher is a tool for defining server clusters and dynamically balancing loads across the cluster according to user-selected rules. Network Dispatcher provides functions for load balancing TCP/IP traffic to designated clusters of Web servers. It is frequently deployed to distribute HTTP requests across Web servers in a cluster, but can also perform the same load balancing function for other TCP/IP services such as telnet, FTP, SMTP, NNTP and others.

Scalability is a key requirement for an enterprise e-commerce site. IBM Network Dispatcher can play a key part here in enabling a flexible cluster of low-end Web server machines to act as a front end to the site. Additional servers can be installed at any time and simply added to the cluster definition on the Dispatcher machine.

Web Traffic Express

Web Traffic Express is a caching proxy Web server. It improves Web site performance by caching static content at the network access boundary, thereby reducing load on the network and on the server which holds the target content. It can be configured as a forward proxy to improve client access to the Internet or as a reverse proxy to improve performance and security at the Web server side of the network.

Web Traffic Express can be configured in conjunction with Network Dispatcher to provide a function known as Content Based Routing (CBR) which allows Network Dispatcher to send requests to specific servers based on factors such as the requested URL.

Network Dispatcher overview

Network Dispatcher has many options for load balancing ranging from simple round-robin distribution to balancing based on server loads (for example, dynamically reported from agents running on the target servers). However, a simple configuration can be set up quickly.

For this redbook, Network Dispatcher was installed on AIX to distribute HTTP requests across two Web servers as shown in Figure 19-14.

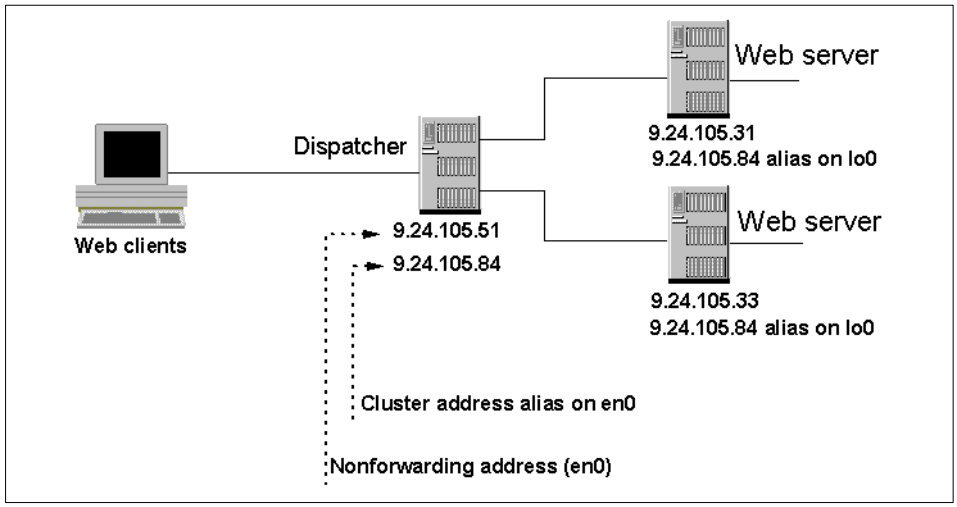


Figure 19-14 Network Dispatcher runtime environment logical view

Web server alias definition

The two Web servers 9.24.105.31 and 9.24.105.33 were installed as part of the 3-tier WCS runtime environment for AIX. Apart from an alias definition required on each machine, no changes were required on these machines for them to be load balanced.

The alias required on each machine is as follows:

```
ifconfig lo0 alias 9.24.105.84 netmask 255.255.255.0
```

This command establishes the IP address of the cluster as an alias on the Web server's loopback adapter. The allows the Web server to accept packets forwarded from the Network Dispatcher, which have the address of the cluster as the destination address.

Additional machines can be added to the cluster at any time. There is no requirement for the target machines to have the same hardware or software environment.

Network Dispatcher alias definition

IBM Network Dispatcher was installed on the AIX with the IP address 9.24.105.51, as seen in Figure 19-14 on page 601. Network Dispatcher was configured to include the two Web servers as a cluster on port 80.

IP address 9.24.105.84 was configured on the Network Dispatcher machine as an alias on the ethernet adapter.

```
ifconfig en0 alias 9.24.105.84 netmask 255.255.255.0
```

This machine now has two addresses:

1. The *cluster address* (9.24.105.84) is the address to which name servers will resolve the Web site URL. It is effectively an alias representing all the machines in the cluster.
2. The *nonforwarding address* (9.24.105.51) is the 'real' address for this machine and is only used for administrative connections (for example, telnet).

Use the fully-qualified host name (for example, clustera.itso.ral.ibm.com). When a client requests a connection to http://clustera.itso.ral.ibm.com, the local name server will resolve this to 9.24.105.84 (the cluster address). Network Dispatcher then has to decide which target server should receive the request.

In the test scenario the Network Dispatcher was initially configured to make this decision based on the number of active and new connections to the servers and on the HTTP response based on periodic test connections. No load-reporting agents were installed on the target servers.

Network Dispatcher will forward the connection request to the chosen Web server. Note that the Web server will see the original client IP address as the origin IP address and thus will respond directly to the client. Responses are *not* routed via the Dispatcher.

Network Dispatcher installation (AIX)

Network Dispatcher was installed on AIX from the WebSphere Edge Server CD. The CD contains installation instructions in the Edge Server *Getting Started* guide and in the Network Dispatcher *User's Guide*. The steps outlined below are condensed from the instructions in the *Getting Started* document.

1. Verify that the machine meets the hardware and software prerequisites for each component and type of functionality you are installing and configuring.
2. Log in to AIX as user ID root, and start a terminal window.
3. Insert the WebSphere Edger Server CD into the CD-ROM drive.

4. Mount the CD as follows:

```
# mount -r -v cdrfs dev/cd0 /mnt
```

5. Start the InstallShield Wizard as follows:

```
# cd /mnt
# ./setup
```

6. Read the license agreement that appears, and if in agreement click **Accept**.
7. On the Select Edge Server Component panel, click the **Network Dispatcher** checkbox.
8. On the Select Network Dispatcher Installation Method panel, click the **Guided** radio button.
9. On the Select Dispatcher Component to Install panel, click the **Dispatcher Component** radio button.
10. Verify that you have satisfied all of the requirements described on the What Should I Do Before I Begin? panel.
11. .On the Define Name of Load-balanced Cluster panel, specify the hostname to assign to the cluster of load-balanced machines. As noted on the panel, clients send their requests to this hostname (or the corresponding IP address) rather than directly to the load-balanced machines, and Network Dispatcher distributes the requests to the load-balanced machines. An example for a cluster of HTTP content hosts at the ABC Corporation's main Web site is `www.abc.com`.
12. .On the Define Port Number of Service to Load Balance panel, specify the port number on which the load-balanced servers accept requests of ascertain protocol type. Either select one of the choices in the upper box or enter the desired port number in the lower box and click on the **Add** button.
13. . On the Define Server Machines to Load Balance panel, specify the machines in the load-balanced cluster, by entering each one's hostname in turn in the **Server to Add** box and clicking the **Add** button. The hostname appears in the upper box. To remove a hostname from the upper box, select it and click the **Remove** button.
14. .On the Complete the Installation panel, verify that the box describes the configuration you have specified. If the InstallShield Wizard discovers that Edge Server software is already installed, it warns you on the Installed Version Detected panel that it is overwriting the existing software.When the installation process completes, check the Installation Summary panel for any error messages. Then read the Edge Server readme.txt file displayed on the Readme Information panel. When you click the **Exit** button on this panel, the InstallShield Wizard window closes. The InstallShield Wizard creates the /usr/lpp/nd directory and installs Dispatcher software and documentation in subdirectories of it.

15. Issue the `ndserver` command to start the Dispatcher server component.
16. If you wish, remove the Edge Server CD-ROM and unmount the CD-ROM drive.

```
# umount <CD_drive>
```

Network Dispatcher administration

Four methods are available for configuring the Network Dispatcher:

1. Command line

   The complete set of `ndcontrol` commands is documented in the *IBM Network Dispatcher User's Guide.* These command allow all aspects of configuration to be performed at the command line.

2. Scripts

   A set of commands can be stored in a script file and executed to configure and run the Dispatcher.

3. Graphical user interface (GUI)

   All aspects of Dispatcher configuration are available using a graphical user interface.

4. Configuration wizard

   The configuration will step through a guided series of panels to prompt for the information necessary to create a basic configuration.

Network Dispatcher administration using the GUI

To start the configuration wizard, if the Network Dispatcher is not already running type `ndserver`. To start the configuration wizard if Network Dispatcher is running type `ndwizard`.

Note: On the Windows NT platform, the Network Dispatcher server service is started automatically as a service.

The Network Dispatcher administration GUI looks like Figure 19-15.

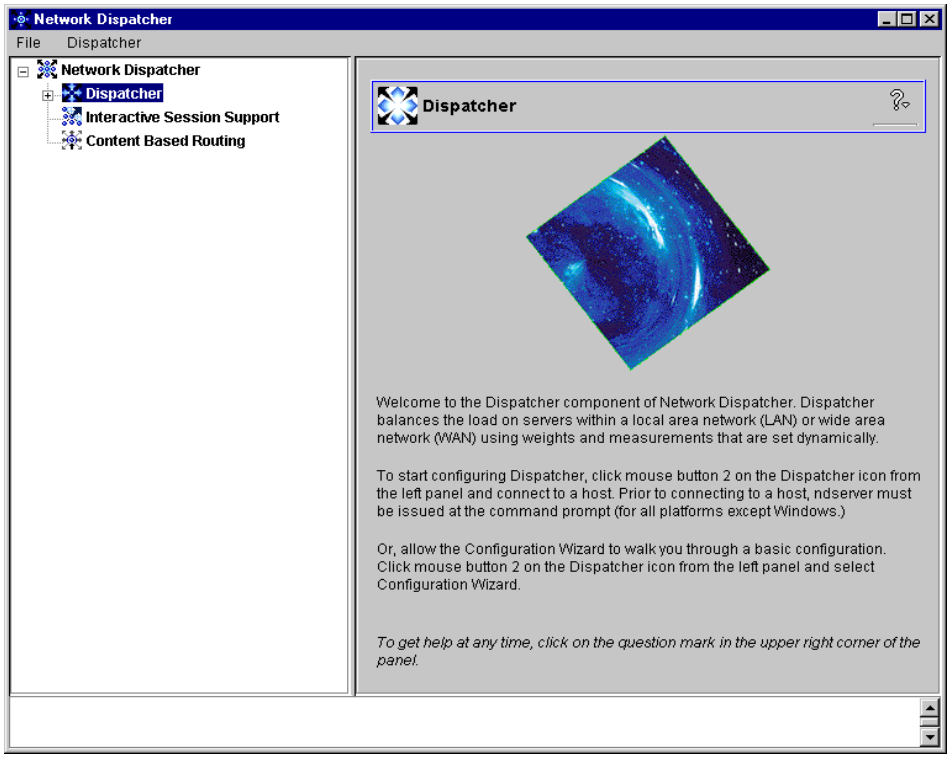


Figure 19-15 Network Dispatcher administration GUI

To start configuration a connection must be made to a host. Unless remote administration is being used, this will be the host on which Network Dispatcher is running. To make this connection right button click on **Dispatcher** and enter the hostname or IP address of the Dispatcher server.

The tree structure shown in the left panel expands to show all the configured components.The panel on the right provides a notebook-style interface in which the parameters for each component can be specified (see Figure 19-16).

At any time the configuration can be saved to a configuration file or a new configuration file loaded. In this example the configuration notebook is open at the HTTP Advisor page. The left panel displays a hierarchical view of the Dispatcher cluster showing the Dispatcher server at 9.24.105.51, the cluster address at 9.24.105.84 and the two servers 9.24.105.31 and 9.24.105.33 being balanced with an HTTP advisor (probe) running on port 80.

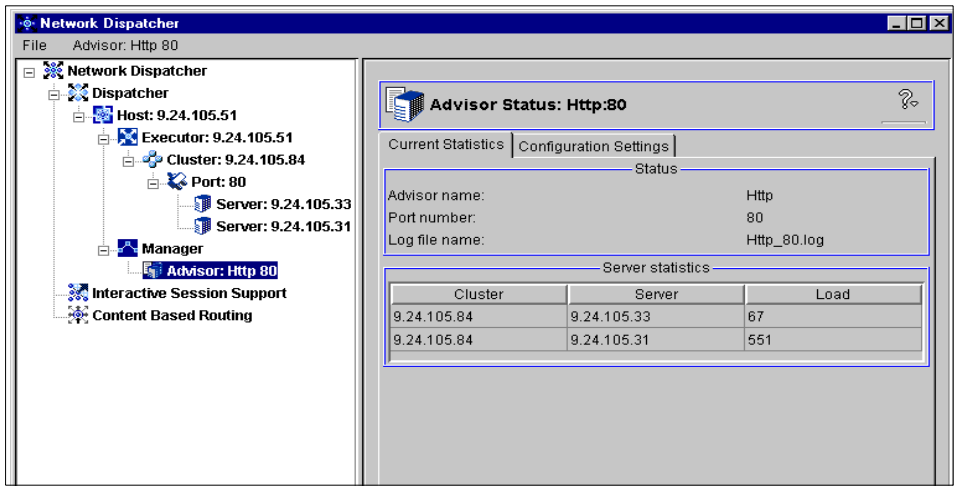


Figure 19-16 Network Dispatcher configuration notebook

Network Dispatcher graphical monitor

The configuration GUI includes a graphical monitor which can display server loads and connection statistics dynamically (see Figure 19-17).

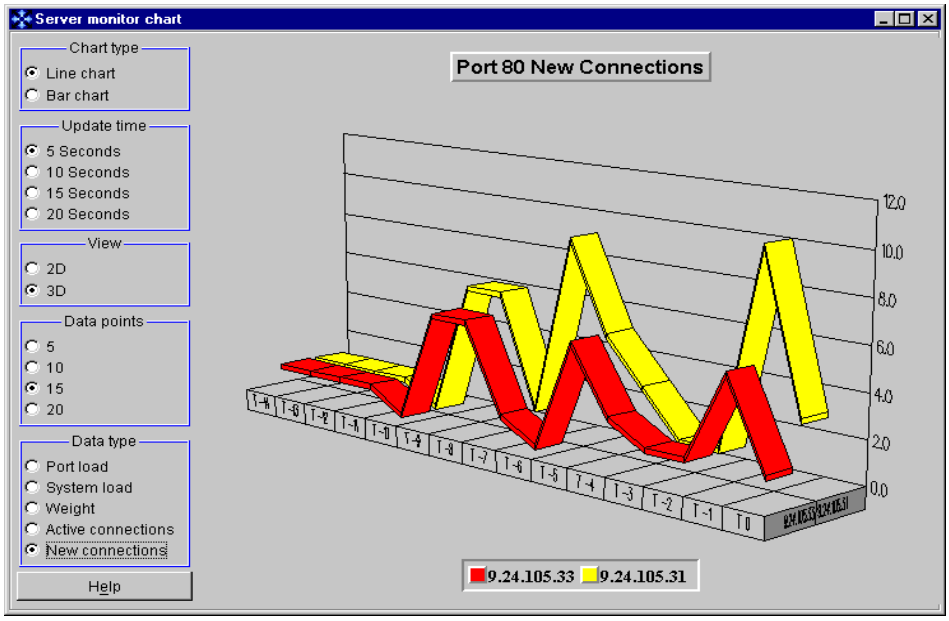


Figure 19-17 Network Dispatcher graphical monitor

Network Dispatcher remote administration

The Network Dispatcher also supports *remote administration* whereby a Network Dispatcher server can be administered from a remote machine acting as an administration client.

This was tested using Network Dispatcher installed on an NT machine to administer the Dispatcher server that had been installed on AIX. To enable remote administration it is first necessary to create authentication keys on the server using the command `ndcreate keys`.

This will create a private key in the /usr/lpp/nd/admin/keys/dispatcher directory. The keyfile will have a name of the form n.n.n.n-10099.key where n.n.n.n is the server IP address and 10099 is the port that is used for RMI communication between the administration client and the server. For the NT client this keyfile needs to be installed in the client\admin\keys\dispatcher directory which is under the directory Dispatcher was installed to.

With this key installed it is possible to connect to the remote Network Dispatcher server and perform configuration tasks. It is also possible to run the graphical monitors remotely.

Session affinity in Network Dispatcher

In some circumstances it may be desirable for the Network Dispatcher to bind a series of HTTP requests from a given client to the same Web server. If the end-user is engaged in a logically related set of requests in which the client state is stored on the server then all subsequent requests should be dispatched to the server which handled the first connection in the session.

- Sticky port affinity

  The basic Network Dispatcher *session affinity* feature allows a ‘stickytime’ to be set for a port. This sets a timer when a transaction is received. If another connection is received from the same client before the timer expires, then the request is dispatched to the same server that was previously selected and the timer is reset.

Note: The client IP address is not always reliable since clients could be connecting via ISP gateways in which large numbers of client connections appear to originate from the same IP address.

- Server directed affinity

Server Directed Affinity (SDA) allows the selection of a server to be controlled by the applications running on the server itself. An agent running on the server can communicate with the Dispatcher sending updates the Dispatcher's affinity tables which can force affinity to the desired server.

- Cookie based affinity

  Network Dispatcher also can implement affinity via the use of cookies (in conjunction with Web Traffic Express and the content based routing feature).

  This allows a server to send an identifying cookie back to the client. When a request is received from the client, the Dispatcher can use the information in the cookie to find the address of server it should select - depending on the rules that are configured for cookie based affinity.

Where to find more information

- For a full description of the IBM WebSphere Edge Server products refer to:

  `http://www.ibm.com/software/Webservers/edgeserver`

- *IBM Network Dispatcher User's Guide* found on the IBM WebSphere Edge Server CD or on the Web at:

  `http://www.ibm.com/software/Webservers/edgeserver/library.html`

- *WebSphere Edge Server: Working with Web Traffic Express and Net Dispatcher*, SG24-6172
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153
- *WebSphere V3.5 Handbook, Using WebSphere Application Server V3.5 Standard and Advanced Editions*, SG24-6161

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 613.

- *Mobile Commerce Solution Guide using WebSphere Commerce Suite V5.1*, SG24-6171
- *WebSphere Commerce Suite V5.1 Customization and Transition Guide*, SG24-6174
- *WCS V5.1 Performance Tuning,* SG24-6258
- *WebSphere V3.5 Handbook*, SG24-6161
- *WebSphere Edge Server: Working with Web Traffic Express and Net Dispatcher*, SG24-6172
- *Programming with VisualAge for Java Version 3.5*, SG24-5264
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755
- *Version 3.5 Self Study Guide: VisualAge for Java and WebSphere Studio*, SG24-6136
- *WebSphere Scalability: WLM and Clustering Using WebSphere Application Server Advanced Edition*, SG24-6153
- *e-Commerce Patterns using WebSphere Commerce Suite, Patterns for e-business Series*, SG24-6156
- *Patterns for e-business: User-to-Business Patterns for Topology 1 and 2 using WebSphere Advanced Edition*, SG24-5864
- *Building e-commerce Solutions with Net.Commerce: A Project Guidebook*, SG24-5417
- *WebSphere Personalization Solutions Guide*, SG24-6214
- *Business-to-Business Integration Using MQSeries and MQSI, Patterns for e-business Series*, SG24-6010
- *IBM Certification Study Guide, AIX V4.3 System Administration*, SG24-5129

© Copyright IBM Corp. 2001

- *AIX Version 4.3 Differences Guide*, SG24-2014-02
- *User-to-Business Patterns Using WebSphere Advanced and MQSI: Patterns for e-business Series*, SG24-6160
- *The XML Files: Using XML for Business-to-Business and Business-to-Consumer Applications*, SG24-6104
- *DB2 UDB V7.1 Performance Tuning Guide*, SG24-6012
- *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163
- *LDAP Implementation Cookbook*, SG24-5110
- *User-to-Business Pattern Using WebSphere Personalization Patterns for e-business Series*, SG24-6213

Other resources

These publications are also relevant as further information sources:

- *Programmer's Guide, IBM WebSphere Commerce Suite V5.1*, product guide found on WebSphere Commerce Suite V5.1 CD.
- *Fundamentals, IBM WebSphere Commerce Suite V5.1*, product guide found on WebSphere Commerce Suite V5.1 CD.
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Windows NT and Windows 2000*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for AIX*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Installation Guide, IBM WebSphere Commerce Studio V5.1 Professional Developer Edition for Windows NT and Windows 2000*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Store Developer: Creating a Store Using the Store Services, IBM WebSphere Commerce Suite V5.1*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Store Developer: Building a Store Archive, IBM WebSphere Commerce Suite V5.1*, product guide found on the WebSphere Commerce Suite V5.1 CD.
- *Installation Guide, IBM WebSphere Commerce Suite V5.1 Pro Edition for Solaris*, product guide found on the WebSphere Commerce Suite V5.1 CD.

- *Administrator's Guide, IBM WebSphere Payment Manager for Multiplatforms V2.2* found on the product CD.
- *Multicultural enablement with WebSphere Commerce Suite*, white paper found at:

  http://www.ibm.com/software/webservers/commerce/wcs_pro/lit.html
- *IBM Network Dispatcher User's Guide* found on the IBM WebSphere Edge Server CD or on the Web at:

  http://www.ibm.com/software/Webservers/edgeserver/library.html
- *IBM SecureWay Directory V3.2 for AIX Installation and Configuration Guide* found on the product CD.
- For more information about WebSphere Commerce Analyzer, refer to the IBM *WebSphere Commerce Analyzer User's Guide*, found on the product CD.
- *Oracle8i Concepts, Release 2 (8.16)*, A76965-01 found at:

  http://otn.oracle.com/docs/products/oracle8i/doc_index.htm
- *Oracle8i Installation Guide*, Release 2 (8.1.6) for Sun SPARC Solaris, Part No. A77181-01 found at:

  http://otn.oracle.com/docs/products/oracle8i/doc_index.htm
- *Oracle8i Administrator's Guide, Release 2 (8.16) for Sun SPARC*, A77184-01 found at:

  http://otn.oracle.com/docs/products/oracle8i/doc_index.htm
- *Net8 Administrator's Guide, Oracle8i Release 8.16*, A76933-01 found at:

  http://otn.oracle.com/docs/products/oracle8i/doc_index.htm
- *Installation and Migration Guide, Netscape iPlanet Web Server, Enterprise Edition V4.1*, 806-4642-01 found at:

  http://developer.iplanet.com/docs/manuals/enterprise.html
- Mark G. Sobell, *A Practical Guide to Solaris*, Addison Wesley, 1999, ISBN 020189548X
- Janice Winsor, *Solaris 7 Reference*, Sun Microsystems Press, A Prentice Hall Title, 1999, ISBN 0130200484

Referenced Web sites

These Web sites are also relevant as further information sources:

WebSphere Commerce Suite

- http://www.ibm.com/software/webservers/commerce/ IBM WebSphere Commerce Suite home page.
- http://www.ibm.com/software/webservers/commerce/servers/versions.html/ IBM WebSphere Commerce Suite version matrix.
- http://www.ibm.com/software/webservers/commerce/servers/support.html/ IBM WebSphere Commerce Suite technical support.
- http://www.ibm.com/software/webservers/commerce/servers/lit-tech.html/ IBM WebSphere Commerce Suite technical library.
- http://www.ibm.com/software/webservers/commerce/servers/downloads.html/ IBM WebSphere Commerce Suite downloads.
- http://www.ibm.com/software/webservers/commerce/community/ IBM WebSphere Commerce Suite community.
- http://www.brio.com/library/ Brio Broadcast Server document library.
- http://www.macromedia.com/software/likeminds/ Macromedia LikeMinds home page.
- http://www.segue.com/ Segue Silk Preview home page.
- http://www.cybercash.com/ CyberCash home page.

IBM Fixpacks and E-fixes download sites

- http://www.ibm.com/software/webservers/appserv/efix.html/ IBM WebSphere Application Server.
- http://www.ibm.com/software/webservers/commerce/wcs_pro/downloads.html IBM WebSphere Commerce Suite V5.1, Pro Edition.
- http://www.ibm.com/software/webservers/studio/support.html/ IBM WebSphere Studio.
- http://www.ibm.com/software/webservers/httpservers/support.html/ IBM HTTP Server.
- http://www.ibm.com/software/ad/vajava/support.htm/ IBM VisualAge for Java.
- http://www.ibm.com/software/data/db2/udb/support.html/ IBM DB2 Universal Database.

Development and general information sites

- http://www.ibm.com/developerworks/patterns/ IBM Patterns for e-business
- http://www.ibm.com/vadd/ IBM VisualAge for Java Developer Domain.
- http://www.ibm.com/developerworks/ IBM DeveloperWorks.

- http://www.ibm.com/software/developer/eeng/ IBM DeveloperWorks for Europe (English)
- http://www.alphaworks.ibm.com/ IBM alphaworks home page
- http://websphere.ibm.com/ IBM WebSphere Central home page
- http://www.ibm.com/websphere/developer/ IBM WebSphere Developer Domain
- http://www.unicode.org/ Information about Unicode UTF-8 encoding.
- http://www.ibm.com/software/Webservers/edgeserver/ Information on IBM WebSphere Edge Server.
- http://www.ibm.com/software/network/directory/ IBM SecureWay Directory home page.

Sun Solaris information

- http://www.sun.com/bigadmin/ General info, patches, documentation, downloads, discussions.
- Solaris 7 documentation found at:
  - http://docs.sun.com/ab2/products_C/INDEX/@ProductViewer/5423/*;td=1?Ab2Lang=C&Ab2Enc=iso-8859-1#CurrentPos/ Solaris 7 documentation
  - http://docs.sun.com/ab2/coll.214.4/@Ab2CollView?Ab2Lang=C&Ab2Enc=iso-8859-1/ Collection of installation information on Solaris 7
- Collection of systems administrator information on Solaris 7 found at:
  - http://docs.sun.com/ab2/coll.47.8/@Ab2CollView?Ab2Lang=C&Ab2Enc=iso-8859-1/ Collection of systems administrator information on Solaris 7.
  - http://www.sun.com/bigadmin/patches/index.html/ Solaris 7 patches

How to get IBM Redbooks

Search for additional Redbooks or redpieces, view, download, or order hardcopy from the Redbooks Web Site

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web Site for information about all the CD-ROMs offered, updates and formats.

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of other companies:

© Copyright IBM Corp. 2001

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

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.

Index

Symbols
.profile update for Oracle8i 230

Numerics
1-tier high level installation steps 68
1-tier runtime configuration 32
2-tier high level installation steps 70
2-tier runtime configuration 33
3-tier enterprise runtime configuration 36
3-tier high level installation steps 73
3-tier runtime configuration 35

A
Accelerator 440
Adapters 40
AddBroadcastJob command 459
AddJob command 457
Address format 378
Administration Console 12, 440
Affinity
  cookie based affinity 608
  server directed affinity 608
  sticky port affinity 607
AIX runtime for WCS 137
Alias for httpd.conf 118, 177
Application design
  model-view-controller 46
Application development 47
  advanced customization 49
  basic customization 48
  intermediate customization 49
  MVC architecture for WCS 47
Architect 17
Auction rules 403
Auction tables 420
Auction wizard 404, 416
Auctions 5, 8, 322
  AutoBids 404
  bidding data model 421
  CleanJob command 420
  create an auction 406, 415
  CreateAuction command 419
  database tables 419
  enable scheduled jobs 405
  overview 402
  sample auction store 406
  scheduled jobs 404
  types of auctions 402
    Dutch auction 402
    open cry auction 402
    sealed bid auction 402
Authentication support 10
AutoBids 404
availability 30

B
B2B 322
B2C 322
Back-end integration 31
Blaze Advisor 7, 287
Business user roles 18
  customer service representative 19
  marketing manager 18
  merchandising manager 18
  merchant 18
  order clerk 19
  project manager 18
Business-to-Business (B2B) 5
Business-to-Consumer (B2C) 5

C
Cache 318
  clearing the cache 370
  configuration for WCS 446
  maintaining the cache 371
  management 450
  optimization 450
  WAS 372
  WCS 371
  Web browser 372
Cache wizard 446
CacheDelete command 451
Caching for WCS 445
Caching parameters for WCS 449
Catalog assets

© Copyright IBM Corp. 2001

store archive 434
Catalog content 378
CLEANCONF table 452
CleanJob command 420, 458
Clearing the cache 370
CMDREG table 51, 53
Command factory 56
Command line
WCS instance creation 80
Command line store deployment 427
Command registry 53
Commands 51, 54
command factory 56
controller command 54
task commands 56
view commands 57
Commerce Studio projects 279
Commerce Suite Accelerator 11
Commerce/WebSphere Application Server 179
Common server runtime 9
Compile WCS admin JSPs 263
CompleteOrder 404
Configuration
AIX
DB2 client 182
IBM HTTP Server 147
IPSec tunneling 207
OSE on the HTTP Server 186
OSE on WAS 185
Payment Manager and WCS 548
SecureWay Directory V3.2 491
WebSphere Application Server 162
Solaris
enable SSL for iPlanet 218
Oracle8i client 243
Oracle8i server configuration 230
Solaris 7 configuration 591
WebSphere Application Server 251
Windows NT/2000
3-tier httpd.conf WCS entries 133
DB2 client-server connectivity 122
DB2 server 99
development environment 300
IBM HTTP Server 92
serve static WCS content 135
VisualAge for Java 302
WebSphere Application Server 106
WebSphere Studio 300
Configuration Manager 80
WCS instance creation 80
configvaj.bat 305
Controller commands 54
Cookie based affinity 608
CreateAuction command 419
Creating a multicultural enabled store 393
Creating an alias 118
Currency 377
Customer service representative 19

D

Data bean manager 57
Data beans 9, 57
data bean manager 57
implementation 57
Data format and representation 377
Database administrator (DBA) 17
Database Configuration Assistant 598
Database creation
WCS manual database creation 80
Database server 39, 179
DB2 Control Center 445
DB2 V7.1
client installation for Windows 121
client installation on AIX 180
client-server configuration for Windows 122
client-server configuration on AIX 182
Fixpack 2a installation for Windows NT/2000 98
server configuration for Windows 99
create WAS database 100
stop DB2 services 99
update JDBC level 99
server install for Windows NT/2000 96
server installation for AIX 151
server pre-installation requirements 152
server verification for Windows NT/2000 98
dbclean command 453
dbclean utility 452
dbora 231
dbstart 231
Debugging methods 60
program trace messages 61
WCS logs 61
WTE 60
Deleting a WCS instance 84
Deploy a SAR
command line 368
Deploy sample store 115, 173, 265

command line 175
Configuration Manager 173
Deploying a SAR 367
Deployment 426
developer runtime environment 426
methods of deployment 426
production runtime environment 426
staging runtime environment 426
test runtime environment 426
Development environment 283
advanced customization tools 285
basic customization tools 285
configuration 300
hardware prerequisites 291
install prerequisites 293
intermediate customization tools 285
ITSO hardware 292
ITSO software 292
methods of installing 290
Commerce Studio umbrella install 290
manual component install 290
shared developer install 290
tips 315
cache 318
disable Payment Manager 316
start PM as a service 318
stop unneeded Windows services 315
tools
HotMedia 289
PerfectPhoto 289
store archive 289
Store Services 288
VisualAge for Java V3.5, EE 287
WebSphere Commerce Studio V3.5 286
WebSphere Commerce Studio V3.5 extensions 286
WebSphere Studio V3.5 286
XML editor 288
tools overview 284
where to find information 319
Directory servers 32
Disable Payment Manager 316
DMT
See SecureWay Directory Management Tool
DoAuctionNotify 404
DTD Generate command 431
DTD Generator 428
Dutch auction 402

E

e-business 4
e-commerce 4, 5
concepts 5
product catalog 5
shopping cart 6
shopping flow 6
user profile 6
models 5
auctions 5
Business-to-Business (B2B) 5
Business-to-Consumer (B2C) 5
Education 21
courses for WAS and VAJ 22
courses for WCS V5.1 21
roadmaps 22
EJBs 44
Enhanced secure payment options 8
Enterprise bean entity framework 9
Enterprise JavaBeans
See EJBs
Error handling 58
Error messages 58, 59
Error reporting 59
e-tailing 4
Exception handling framework 58
Export Store Archive 287
Extensible Markup Language
See XML

F

File transfer deployment 427
FinalizeAuction 404
Firewalls 32

G

getResource.jsp 392
GSKit 503
GSKit installation 504

H

Horizontal scaling 30
HotMedia 289
HTTP Server Administration Console 444
httpd.conf entries for 3-tier 198

I

I/T specialist roles 16
  architect 17
  database administrator 17
  Java programmers 17
  site administrator 17
  store administrator 17
  Web designers 17
IBM Global Security Kit
  See GSKit
IBM HTTP Server
  configuration for AIX 147
  configuration for Windows NT/2000 92
    create HTTP server admin user 94
    enable SSL 92
  installation for Window NT/2000 91
  installation on AIX 144
  logging 463
  verification for Windows NT/2000 95
IBM Learning Services 21
IBM XML Tools 288, 299
i-commerce 4
id command 594
ID resolver 428
IDResolve command 432
Import Store Archive 287
Inbound messages 555
Installation
  AIX
    1-tier runtime 139
    3-tier runtime 179
    AIX file sets and service prereqs 142
    AIX install 140
    DB2 client 180
    DB2 server 151
    IBM HTTP Server 144
    Network Dispatcher 602
    SecureWay Directory V3.2 480
    WAS V3.5 157
    WAS V3.5 Fixpack 2 160
    WAS V3.5.2 E-fixes 161
    WebSphere Commerce Suite 164
    WebSphere Payment Manager 540
    WebSphere plug-in 183
  development environment 291
  development environment prerequisites 293
  Solaris
    iPlanet Web Server 213
    Netscape Communicator 214
    Oracle8i client 241
    Oracle8i Server 222
    Solaris 7 installation 586
    WAS V3.5 245
    WAS V3.5 Fixpack 2 254
    WAS V3.5.2 E-fixes 254
    WebSphere Commerce Suite 256
  WCS installation methodologies 66
  Windows NT/2000
    1-tier runtime 88
    2-tier runtime 120
    3-tier runtime 126
    DB2 Client 121
    DB2 Fixpack 2a 98
    DB2 Server 96
    IBM HTTP Server 91
    VisualAge for Java 295
    WAS V3.5 101
    WAS V3.5 Fixpack 2 103
    WAS V3.5.2 E-fixes 104
    WebSphere Commerce Studio 298
    WebSphere Commerce Suite 108
    WebSphere plug-in 126
    WebSphere Studio 293
Instance
  multiple WCS instances 81, 84
  WCS instance creation 77
  WCS instance overview 77
iPlanet Web Server 213
  enable SSL 218
  installation on Solaris 213
  pre-installation requirements 215
  verification 221
IPSec tunneling 206
  AIX prerequisites 206
  configuration on AIX 207
  starting 208

J

J2EE 44
Java 2 Platform, Enterprise Edition
  See J2EE
Java Database Connectivity
  See JDBC
Java programmer 17
Java server programming 44
  EJBs 44
  JDBC 45

JSPs 45
servlets 45
XML 45
Java Servlets 45
See servlets
java.security 106
JavaServer Pages
See JSPs
JavaServer Pages (JSPs) 9
JDBC 45
Job scheduler tables 460
Job scheduler utility 456
AddBroadcastJob command 459
AddJob command 457
CleanJob command 457
RemoveJob command 458
JSPs 45, 58, 382

L

Language description table 396
Language support in WCS 376
LDAP 471
directory 472
overview 472
usage 473
LDAP authentication 10
LDAP design features in WCS 476
LDAP limitations in WCS 477
LDIF 499
Lightweight Directory Access Protocol
See LDAP
Load balancers 32
Load command 433
Loader package 12, 189, 428
copy files 189
data load process 428
sample data load 430
usage considerations 434
zip files 189
zip up files 81
Loader utility 428
Logging 463
IBM HTTP Server 463
WAS 463
WCS 464

M

MA88 561
Machine A
Web Server 179
Machine B
Commerce/WebSphere Application Server 179
Machine C
Database Server 179
Macromedia LikeMinds 7
Marketing campaigns 7
Marketing manager 18
m-commerce 4, 9
Member and group management 8
Merchandising manager 18
Merchant 18
Message transports 554
Messaging and queueing 554
Microsoft Internet Explorer V5.5 293
mkdir command 593
Mobile commerce 9
Mobile devices 4
mobile phone 4
wireless laptop 4
wireless PDA 4
Mobile phone 4
Model-view-controller 46
MonitorAuctions 404
MQSeries 553
classes for Java 561
classes for Java Message Service 561
MQSeries adapter 558, 566
Multicultural enablement 375
JSPs 382
Multicultural enablement in WCS 380
approach 381
JSPs 382
language ID and locale 384
properties files 382
resource bundles 384
sarinfo.xml 386
template management 386
tools 380
unicode 385
UTF-8 encoding 385
Multicultural overview 376
Multicultural parts of store 376
address format 378
catalog content 378
currency 377
data format 377
language 376

page design 378
payment methods 379
pricing 378
shipping 379
taxation 379
Multicultural programming model 388
Multicultural support 7
Multilingual static text 389

N

National language 376
Net8 232, 597
Netscape Communicator for Solaris 214
Netscape Directory 475
Netscape iPlanet Web Server 213
Network Dispatcher
administration 604
graphical monitor 606
overview 600
remote administration 607
session affinity 607
Web server alias 601
where to find information 608
Network Dispatcher installation 602

O

Online catalog 7
Open cry auction 402
Oracel8i
where to find information 598
Oracle8i
check status of listener 597
client configuration 243
client installation on Solaris 241
commands and tasks 596
configure database autostart 235, 236
create database 233
create oracle ID and tablespace 236, 237
create user account 226
Database Configuration Assistant 598
database tuning 235
database verification 235
dbora 231
dbstart 231
kernel configuration 225
listener 596
Net8 232
Configuration Assistant 597
Net8 configuration 239
prepare databases for WAS, WCS 233
server installation on Solaris 222
start listener 596
start server 596
stop listener 597
user .profile update 230
Oralce8i
stop server 596
Order clerk 19
Order processing 8
OSE
configuration on the HTTP Server 129, 186
configuration on WAS 128, 185
Outbound messages 556

P

Page design 378
Payment Manager
disable WCS integration 316
start as Windows service 318
payment methods 379
PerfectPhoto 289
Personalization 7
Pricing 378
ProcessAutoBids 404
ProcessDutchBids 404
ProcessOpenCryBids 404
Product packaging for WCS 12
development product packaging 12
runtime product packaging 12
Programming model
concepts 44
multicultural support 388
Project manager 18
Properties files 382
Protocol listeners 40
ps command 593
Publish from Store Services 115, 173
Publishing a SAR 278
publishstore.bat 368

R

Redbooks Web Site 613
Contact us xx
RemoveJob command 458
Resource bundles 384
Roles and skills

16
business user roles 18
I/T specialist roles 16
Runtime architecture 37
software components 38
Database server 39
Web server 39
WebSphere Application Server 39
WebSphere Commerce Server 39
WCS components 39
adapters 40
protocol listeners 40
Servlet Engine 40
Web Controller 40
Runtime computing flow 41
Runtime environment
1-tier configuration 32
1-tier high level installation steps 68
2-tier configuration 33
2-tier high level installation steps 70
3-tier configuration 35
3-tier high level installation steps 73
AIX 138
availability 30
back-end integration 31
business considerations 30
computing flow 41
configurations 32
enterprise 3-tier configuration 36
horizontal scaling 30
installation methodologies 66
scalability 30
security 31
skills 31
vertical scaling 30
Windows NT/2000 88
hardware and software prerequisites 89
installation and service packs 90
ITSO hardware 88, 89
ITSO software 90
overview 88

S

Sample store URL alias 118
SAR 272, 275
Commerce Studio data flow 280
create PackageSAR directory 357
create publishing server 358
create SAR packaging stage 358
creating a SAR 277
creating a store archive 355
creation overview 356
define assets to publish 359
deploy from command line 368
deploy from Store Services 367
deployment 367
high level creation steps 356
high level packaging procedure 360
import assets in Studio 357
multicultural store 276
non-multicultural store 276
package command line batch files 361
package via command line 360
package via GU 364
packaging a SAR 359
publish to PackageSAR directory 359
structure 275
unzip to directory 329
sarinfo.xml 386
Scalability 30
Sealed bid auction 402
SecureWay Directory
add a suffix 497
add entries to database 499
administration 496
configuration on AIX 491
delete a suffix 498
installation on AIX 480
SSL configuration 502
SecureWay Directory Management Tool 516
SecureWay Directory V3.2 473
working example 525
Security 31
Serve static content for 3-tier 202
Server directed affinity 607
Servlet Engine 40
Shipping 379
Shutdown 593
Site administrator 17
Solaris
common tasks and commands 593
configuration 591
prereqs for WCS 586
Solaris 7 configuration 586
Solaris 7 installation 586
Solaris 7 patches for WCS 591
where to find information 594

Solaris runtime for WCS 212
Sticky port affinity 607
Store 11
Store administrator 17
Store archive 272
  catalog assets 434
  contents 272
  creating a SAR 355
  database assets 273
  descriptor 273
  file assets 272
  overview 272
  properties resource bundles 273
  publishing 278
  Store Services 277
  Web assets 272
Store Archive Publish Preferences 287
Store Archive template 287
Store archives
  Commerce Studio 279
Store customization
  advanced 20
  basic 19
  intermediate 20
Store deployment 426
Store development
  approaches for creating stores 323
  create a store 323
    create store template 323
    generate store from template 326
    publish a new store 327
    update VAJ WTE 327
  create and customize 321
  customization
    change a descriptor file 351
    change a properties file 350
    change label 349
    change store logo 348
    Web assets 347
  multicultural enablement 375
  multicultural sample 391
  overview 322
  store verification 370
  testing the store 372
  where to find information 352
Store Services 11, 439
Store Services store deployment 427
Systems management 9

T

Tail 593
Task commands 56
Taxation 379
Template management 386
  1 template for all stores 387
  1 template per language 387
  1 template per store 388
Test environments 318
  PC Web browser 319
  VAJ WTE + WCS 319
  WCS V5.1 runtime environment 319
Testing a store 372
Tips for development 315
Trace messages 61
Tracefile 584
Translation of assets 396
Translation of properties files 396
Transports 567
Troubleshooting WCS 466

U

Unicode 385
URL registry 52
URLREG table 51, 52
UTF-8 385

V

Verification
  AIX
    WebSphere Application Server 164
  Solaris
    iPlanet Web Server 221
    WebSphere Application Server 252
  Windows NT/2000
    DB2 server 98
    IBM HTTP Server 95
    VisualAge for Java 297
    WebSphere Application Server 108
    WebSphere Studio 294
Vertical scaling 30
View registry 53
VIEWREG table 53
Views 57
VisualAge for Java 11, 287
  add features 303
  configuration 302
  configure EJB and PNS servers 305

configure the EJB server 308
configure the PNS 309
import wcs5101.dat 303
install verification 297
installation 295
patch 2 install 297
run configvaj.bat 305
start EJB server 311
start Servlet Engine 313
WTE 311
WTE for WCS stores 327

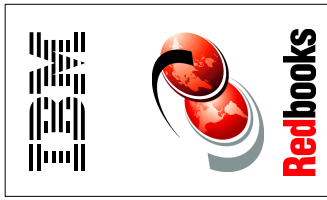
W

WAS
See WebSphere Application Server
WCS
See WebSphere Commerce Suite
WCS instance 77
command line 80
Configuration Manager 80
create on AIX 166
create on AIX via command line 169
create on Solaris 258
create using Configuration Manager 111
creating multiple instances 81
creation methods 77
Command line 77
Configuration Manager 77
creation process 77
deleting a WCS instance 84
manual instance creation 191
multiple WCS instances 84
remote database configuration 124
verify a WCS instance 114
wcs5101.dat 303
Web assets 272
Web Controller 50
Web controller 40
Web designer 17
Web server 39, 179
Web Traffic Express 600
WebSphere Application Server 39
Administration Console 445
configuration on AIX 162
configuration on Solaris 251
configuration on Windows NT/2000 106
configure iPlanet Web Server 249
create WAS database on AIX 157
creating a WAS database 584
installation on AIX 157
installation on Solaris 245
installation on Windows NT/2000 101
logging 463
plug-in install 127
tracefile 584
verify on AIX 164
WebSphere plug-in 183
WebSphere Commerce Analyzer 443
WebSphere Commerce Server 39
WebSphere Commerce Studio 284, 286
install prerequisites 298
installation 298
load store assets 328
VisualAge for Java V3.5, EE 284
WebSphere Commerce Studio extensions 284
WebSphere Studio V3.5, AE 284
XML editor 284
WebSphere Commerce Studio deployment 427
WebSphere Commerce Suite
admin tools 438
Accelerator 440
Administration Console 440
Configuration Manager 439
DB2 Control Center 445
documentation 438
HTTP Server Administration Console 444
Store Services 439
WAS Administration Console 445
WebSphere Commerce Analyzer 444
administrative tools 438
business enhancements 6
auctions 8
enhanced secure payment options 8
marketing campaigns 7
member and group management 8
multicultural support 7
online catalog 7
order processing 8
personalization 7
commands 54
configuration for LDAP 509
course for WCS V5.1 21
data beans 57
development environment 283
development tools 10
education 21
error messages 58

exception handling 58
functionality in V5.1 6
installation on AIX 164
installation on Solaris 256
installation on Windows NT/2000 108
LDAP support 475
logging 464
logging and troubleshooting 460, 462
management and administration tools
Administration Console 12
Commerce Suite Accelerator 11
Loader package 12
Store Services 11
manual database creation 190
messaging 554
overview 4
product packaging 12
skills planning 15
store archive 272
supported LDAP directories 475
technical enhancements 8
common server runtime 9
messaging system 10
mobile commerce 9
multiple authentication methods 10
multiple database support 11
multiple Web server support 11
views 57
Web Controller 50
WebSphere Commerce Suite logs 61
WebSphere Edge Server 600
Network Dispatcher 600
Web Traffic Express 600
WebSphere Payment Manager
configuration for WCS 548
installation on AIX 540
WebSphere Personalization 7
WebSphere plug-in 183
WebSphere Studio 11
configuration 300
configuration after import of assets 336
configure for publishing 338
create project 330
create publishing stages 339
create server to publish to 340
define publishing for SAR 340
define publishing for WCS 342
define publishing for WTE 344
import store assets 331
create directory structure 332
import catalog data assets 335
import properties assets 334
import sarinfo.xml descriptor 335
import Web assets 333
install verification 294
installation 293
publishing assets 346
XML tools registration 336
WebSphere Studio V3.5, AE 286
WebSphere Test Environment
See WTE
WebSphere Test Environment (WTE) 319
Windows NT/2000 runtime for WCS 87
Wireless laptop 4
Wireless PDA 4
WTE 60

X

X.509 authentication 10
XML 45
XML Spy 288, 300
xmlconfig.dtd 106



WebSphere Commerce Suite V5.1 Handbook

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

WebSphere Commerce Suite V5.1 Handbook



Implement and manage a multi-tier commerce runtime environment

Create and customize a store, auctions, data management

Integrate LDAP, Payment, MQSeries

This redbook provides architects, specialists, and developers with the knowledge to develop, implement, and manage stores using WebSphere Commerce Suite V5.1 and WebSphere Commerce Studio V5.1.

This book describes the WebSphere Commerce Suite V5.1 key concepts and features, systems architecture, and programming model, and the skills required to use the Suite.

Detailed examples are provided for implementing a multi-tier WCS runtime environment on the Windows NT/2000 (IHS, DB2), AIX (IHS, DB2), and Solaris (iPlanet, Oracle) platforms.

Included are procedures for installing the development environment, and how to create and customize a store by examples using the WebSphere Commerce Studio V5.1 (VAJ, Studio). Additional examples are provided for creating a SAR, integrating auctions and multicultural support.

Next, we describe the tools and tasks for store deployment, and systems and catalog management. Back-end integration examples are provided for SecureWay Directory (LDAP), Payment Manager, and MQSeries.

The appendixes includes tips on the WebSphere Application Server, Solaris, Oracle, and Network Dispatcher.

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

SG24-6167-00 ISBN 0738422126