Content Manager Implementation and Migration Cookbook

April 2006

This edition applies to Version 8, Release 3 of IBM Content Manager for Multiplatforms (product number 5724-B19); and Version 8, Release 2 of IBM Content Manager for z/OS (product number 5697-H60).

© Copyright International Business Machines Corporation 2004, 2006. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# Contents

## Notices
Notices only appears on this page.

## Trademarks
Trademarks only appears on this page.

## Preface
The team that wrote this redbook.

## Become a published author

## Comments welcome

## Summary of changes


## Part 1. Introduction

### Chapter 1. Content Management products overview

1.1 Content management portfolio

1.1.1 IBM DB2 Content Manager

1.1.2 IBM DB2 Content Manager OnDemand

1.1.3 IBM DB2 Content Manager VideoCharger

1.1.4 IBM DB2 Content Manager CommonStore

1.1.5 IBM DB2 Document Manager

1.1.6 IBM DB2 Records Manager

1.1.7 IBM WebSphere Information Integrator Content Edition

1.1.8 IBM Workplace Web Content Management

1.2 Choosing the right solution

1.2.1 Content Manager

1.2.2 OnDemand

1.2.3 Document Manager

1.2.4 CommonStore

1.2.5 WebSphere Information Integrator for Content Edition

1.2.6 Workplace Web Content Manager

1.3 What’s changed in Content Manager V8.3

### Chapter 2. Content Manager architecture overview

2.1 Architecture

2.2 Content Manager components

2.2.1 Library Server

2.2.2 Resource Manager

2.2.3 Content Manager clients

2.3 Architecture extensions
## Part 2. Understanding the product

### Chapter 3. Data modeling
3.1 Data modeling entities
   - 3.1.1 Attributes
   - 3.1.2 Attribute groups
   - 3.1.3 Components
   - 3.1.4 Objects
   - 3.1.5 Item type
   - 3.1.6 Items
   - 3.1.7 Semantic type
   - 3.1.8 Relationships (links, auto-linking, references, foreign keys)
   - 3.1.9 Database indexes
   - 3.1.10 Text indexes
3.2 The Content Manager meta model
3.3 Comparison with earlier versions
   - 3.3.1 Hierarchical item type
   - 3.3.2 Items
   - 3.3.3 Versioning
   - 3.3.4 Links
   - 3.3.5 References
   - 3.3.6 Attribute groups

### Chapter 4. Workflow
4.1 Workflow options
   - 4.1.1 Document Routing
   - 4.1.2 WebSphere Process Server (WPS)
4.2 Document Routing concepts
   - 4.2.1 Work packages
   - 4.2.2 Processes
   - 4.2.3 Defining a process in the graphical builder
   - 4.2.4 Updating a process
   - 4.2.5 Deleting a process
   - 4.2.6 Worklists
   - 4.2.7 Actions
   - 4.2.8 Action lists
   - 4.2.9 Customization options

### Chapter 5. Text indexing and searching
5.1 Implementation
   - 5.1.1 Enabling text search
   - 5.1.2 Making documents text-searchable
   - 5.1.3 Making attributes text-searchable
   - 5.1.4 Defining text search options
### Table of Contents

5.1.5 Making documents text searchable on Unicode databases ........................................ 127
5.2 Administration .................................................................................................................. 127
  5.2.1 Updating the index .................................................................................................... 127
  5.2.2 Reorganizing the index ............................................................................................. 128
5.3 Using text search. ............................................................................................................. 129
  5.3.1 Searching for object contents .................................................................................. 129
  5.3.2 Searching for documents ......................................................................................... 129
  5.3.3 Making user-defined attributes text-searchable ...................................................... 130
5.4 Performance considerations ............................................................................................. 130

**Chapter 6. Application development overview** .................................................................. 131
6.1 Getting started. ............................................................................................................... 132
  6.1.1 Where the APIs fit in ............................................................................................... 132
  6.1.2 Installing connectors ............................................................................................... 133
  6.1.3 Setting up your environment .................................................................................. 135
  6.1.4 Setting up WebSphere Studio Application Developer ............................................ 137
  6.1.5 Working with sample code ...................................................................................... 139
  6.1.6 Application development options .......................................................................... 141
  6.1.7 Understanding the differences ................................................................................ 143

6.2 Application development concepts ................................................................................ 144
  6.2.1 Understanding components .................................................................................... 144
  6.2.2 Representing items using Dynamic Data Objects (DDO) ...................................... 148
  6.2.3 Working with Resource Manager .......................................................................... 149
  6.2.4 Working with transactions ...................................................................................... 150
  6.2.5 Using logging and tracing ....................................................................................... 154

6.3 Additional resources ....................................................................................................... 160

**Chapter 7. Query language** ............................................................................................ 161
7.1 Query language overview ............................................................................................... 162
  7.1.1 Parametric search ................................................................................................. 163
  7.1.2 Text search ........................................................................................................... 164

7.2 Understanding query language ....................................................................................... 164
  7.2.1 Symbols ................................................................................................................ 164
  7.2.2 Grammar ............................................................................................................... 165
  7.2.3 Escape sequences .................................................................................................. 165

7.3 Query strings .................................................................................................................. 172
  7.3.1 Basics .................................................................................................................... 173
  7.3.2 Multiple item types ............................................................................................... 174
  7.3.3 Text search ............................................................................................................ 174
  7.3.4 Links ...................................................................................................................... 176
  7.3.5 References ............................................................................................................ 177
  7.3.6 Versions ................................................................................................................ 178
  7.3.7 System-defined attributes ...................................................................................... 178
7.3.8 Resource items ........................................................................ 178
7.3.9 Document parts ...................................................................... 179
7.3.10 Lists ..................................................................................... 179
7.3.11 Attribute groups .................................................................... 180
7.3.12 Set operators .......................................................................... 180
7.3.13 Row-based view filtering ......................................................... 181
7.3.14 Query on checked-out items .................................................... 182
7.4 Using query language ................................................................ 182
  7.4.1 Query string .......................................................................... 182
  7.4.2 Query options ....................................................................... 183
  7.4.3 Query results ......................................................................... 184
7.5 SQL queries ............................................................................... 185
7.6 Other resources .......................................................................... 185

Chapter 8. Security ................................................................. 187
  8.1 Content Manager security concepts ............................................ 188
    8.1.1 Content Manager security .................................................. 188
  8.2 Authentication .......................................................................... 189
    8.2.1 LDAP integration overview ................................................. 190
    8.2.2 Single sign-on .................................................................... 192
  8.3 Authorization ............................................................................ 195
    8.3.1 Privileges .......................................................................... 196
    8.3.2 Users and user groups ......................................................... 198
    8.3.3 Creating user IDs and passwords ........................................... 200
    8.3.4 DB2 administration authority ............................................... 201
    8.3.5 Changing password to Resource Manager .......................... 203
    8.3.6 Changing database access passwords ................................. 205
    8.3.7 Access control lists ............................................................ 206
    8.3.8 Access control list user exits ................................................ 212
    8.3.9 Domains ............................................................................ 214
  8.4 Access to objects ...................................................................... 218
  8.5 WebSphere global security ........................................................ 223
    8.5.1 Java 2 security ................................................................... 228
  8.6 Content Manager and RACF ..................................................... 229
    8.6.1 User IDs ............................................................................ 229
    8.6.2 RACF user exit ................................................................... 232
    8.6.3 RACF import utility ............................................................ 233

Chapter 9. Tivoli Storage Manager for Content Manager ............. 235
  9.1 IBM Tivoli Storage Manager ..................................................... 236
    9.1.1 Overview of Tivoli Storage Manager capabilities ................ 236
    9.1.2 Architecture ..................................................................... 238
    9.1.3 Policy objects .................................................................... 242
9.1.4 Storage devices and media ........................................ 245
9.1.5 Storage hierarchy and data migration .......................... 247
9.1.6 Media management .................................................. 248
9.2 Tivoli Storage Manager and Content Manager .................. 249
  9.2.1 TSM server configuration ....................................... 252
  9.2.2 Customizing the TSM API client files ........................... 252
  9.2.3 Configuring a Resource Manager to use TSM ................. 253

Chapter 10. XML support ...................................................... 257
10.1 Introduction ............................................................ 258
10.2 How XML services work with other Content Manager programming layers 258
10.3 Working with Web services .......................................... 259
  10.3.1 Web services overview .......................................... 260
  10.3.2 Content Manager Web services implementation .............. 263
  10.3.3 Integrating basic Web services into your applications or processes .... 264
  10.3.4 Exporting item types to a WSDL file .......................... 268
  10.3.5 Exporting a process as XML Text (Workflow) ................. 271
10.4 XML JavaBeans .......................................................... 273
10.5 XML schema mapping utility ....................................... 276
  10.5.1 Supported scenarios ............................................. 277
  10.5.2 Creating an XML schema file ................................... 277
  10.5.3 Mapping a user-defined schema to a storage schema .......... 279

Part 3. Content Manager implementation .................................. 283

Chapter 11. Planning and designing ....................................... 285
11.1 Planning basics ....................................................... 286
11.2 Analyze business operations and requirements .................... 286
11.3 Planning and designing system topology .......................... 288
11.4 Planning and designing data model ................................ 291
11.5 Planning and designing workflow ................................... 292
  11.5.1 Hardware and software requirements .......................... 293
11.6 Capacity planning ..................................................... 294
  11.6.1 Library Server capacity ......................................... 294
  11.6.2 Resource Manager capacity ...................................... 294
11.7 Planning for performance ............................................. 295
  11.7.1 LAN cache .......................................................... 296
  11.7.2 Replication .......................................................... 297
11.8 Planning and designing text search .................................. 298
11.9 Planning and designing security ..................................... 298
11.10 Options checklist ...................................................... 299
11.11 Summary ............................................................... 299
Chapter 12. Deployment ................................................................. 301
12.1 System verification ............................................................. 302
    12.1.1 Verification utilities ....................................................... 302
    12.1.2 Verify individual components ....................................... 303
    12.1.3 Post-installation changes ............................................. 305
12.2 Deploying custom applications ........................................... 306
12.3 Testing .............................................................................. 307
12.4 Software configuration management ...................................... 308
12.5 Production rollout .............................................................. 308
12.6 Updates ........................................................................... 309

Chapter 13. Case study ................................................................. 311
13.1 Introduction ......................................................................... 312
13.2 Business problem ............................................................... 312
    13.2.1 Requirements ............................................................... 313
13.3 Designing the solution .......................................................... 317
    13.3.1 Architecture ................................................................. 318
    13.3.2 Data model ................................................................. 322
    13.3.3 Workflow ................................................................. 330
    13.3.4 Migration and document life-cycle ................................. 331
    13.3.5 Security .................................................................... 332
    13.3.6 Client ..................................................................... 334
13.4 Implementing the solution ...................................................... 336
    13.4.1 System installation and configuration ............................... 336
    13.4.2 Workflow .................................................................. 340
    13.4.3 Migration and document life-cycle ................................. 341
    13.4.4 Security .................................................................... 341
    13.4.5 Data model .................................................................. 342
    13.4.6 Client ..................................................................... 344

Part 4. Content Manager migration .................................................. 345

Chapter 14. Upgrade and migration on multiplatforms ....................... 347
14.1 Introduction ......................................................................... 348
14.2 Upgrade considerations ........................................................ 352
    14.2.1 General considerations .................................................. 352
    14.2.2 UNIX considerations ..................................................... 355
14.3 Migration considerations ....................................................... 357
    14.3.1 What is new in the migration tool for Content Manager V8.3 357
    14.3.2 Disk space .................................................................. 358
    14.3.3 Backup ..................................................................... 359
    14.3.4 Timing ..................................................................... 360
    14.3.5 Migrating Content Manager systems earlier than Version 7.1 360
14.4 Data migration overview ........................................................ 360
Chapter 17. Application migration .......................... 451
  17.1 Introduction ........................................... 452
    17.1.1 Porting process .................................. 452
    17.1.2 What’s new in V8.3 APIs from V7 ............... 453
  17.2 Application porting scenarios overview .......... 455
  17.3 Information Integrator for Content Java beans ... 457
  17.4 Information Integrator for Content Federated Connector ... 458
  17.5 Information Integrator for Content DL Connector ... 459
  17.6 CM Folder Manager and Library Client API ....... 460
  17.7 Text Search Engine (TSE) .......................... 460
  17.8 Image Search Engine (QBIC) ......................... 461
  17.9 Client for Windows Automation Interface .......... 461
  17.10 Information Mining based applications ......... 463
  17.11 OLE API based applications ....................... 463
  17.12 APIs other than FM APIs not carried into CM V8 .. 463
  17.13 Programming tips ................................ 464
    17.13.1 Packaging for the Java environment ........... 464
    17.13.2 Programming using Content Manager V8 ....... 465
    17.13.3 Working with the Content Manager samples .... 467

Part 5. Maintenance .......................................... 469

Chapter 18. Maintenance ..................................... 471
  18.1 Maintenance tasks overview ....................... 472
  18.2 Optimizing server databases ...................... 472
  18.3 Monitoring LBOSDATA directory size .............. 477
  18.4 Managing staging directory space ................ 480
    18.4.1 Purger process ................................ 481
  18.5 Removing entries from the events table ......... 483
  18.6 Removing log files ................................ 484
  18.7 Managing Resource Manager utilities and services .. 485
    18.7.1 Configuration of Resource Manager utilities and services .. 485
    18.7.2 Configuring the Resource Manager services on UNIX .... 489
    18.7.3 Starting and stopping resource services on UNIX .... 490
    18.7.4 Asynchronous Recovery utility .................. 490
    18.7.5 Validation utilities overview .................. 493
    18.7.6 Resource Manager/Library Server validation utility .. 494
    18.7.7 Resource Manager volume validation utility .......... 497
  18.8 Replacing or repartitioning a hard disk ........ 499
    18.8.1 Replacing the staging volume for UNIX ...... 499
    18.8.2 Replacing the storage volume for UNIX .......... 500
    18.8.3 Replacing the staging volume for Windows ... 501
Chapter 19. Export and import utilities ................................................. 509
  19.1 XML export and import in system administration ......................... 510
  19.2 Exporting data as XML .......................................................... 510
  19.3 Importing data as XML .......................................................... 516
    19.3.1 Understanding the Import Preprocessor Results window ............ 519
    19.3.2 Additional details on each state .................................... 520
    19.3.3 Using the Details window ............................................. 524
    19.3.4 Completing the import .................................................. 525
    19.3.5 Importing a process from XML text ................................ 527
  19.4 Importing and exporting metadata using XML services APIs ............ 530
    19.4.1 Importing and exporting administration objects as XML .......... 532
    19.4.2 Importing and exporting Content Manager data model objects as XML
              schema files (XSD) ..................................................... 533
    19.4.3 Unsupported XML types in the Content Manager storage schemas . 536
    19.4.4 Constraints for converting to Content Manager storage schemas 537
    19.4.5 Importing and exporting Content Manager data instance objects as
              XML ................................................................. 537
    19.4.6 Exporting Content Manager DDO items as XML items ............... 538
    19.4.7 Importing XML items as Content Manager DDO items ............... 540
    19.4.8 Importing and exporting XML object dependencies ................ 541
    19.4.9 Extracting content from different XML sources ..................... 542

Chapter 20. Performance tuning .......................................................... 543
  20.1 Performance tuning basics .................................................... 544
    20.1.1 Understanding performance goals ................................... 544
    20.1.2 General performance tuning guidelines .............................. 545
    20.1.3 Content Manager components ........................................ 546
  20.2 Performance monitoring ....................................................... 546
  20.3 Performance tuning ............................................................. 549
    20.3.1 Tuning Windows .......................................................... 549
    20.3.2 Tuning AIX ............................................................... 550
    20.3.3 Tuning DB2 ............................................................... 550
    20.3.4 Use multiple disks ...................................................... 550
    20.3.5 Customize Content Manager database installation .................. 550
    20.3.6 Separate database instances ......................................... 550
### 20.3.7 Create attribute indexes ........................................ 551
### 20.3.8 Routine runstats/rebind ....................................... 551
### 20.3.9 Tuning WebSphere Application Server ........................ 551
### 20.3.10 Tuning Content Manager .................................... 552

#### Chapter 20.4 DB2 Performance Expert for Content Manager .... 553
#### Chapter 20.5 New features for CM C++ API performance ......... 555
#### Chapter 20.6 Additional resources ............................... 556

### Chapter 21. Troubleshooting ..................................... 559

#### 21.1 Log and trace ................................................... 560
    - 21.1.1 Single logging directory .................................. 560
    - 21.1.2 Consolidated configuration files related to log settings . 561
    - 21.1.3 Configure log through System Administration Client .......... 562
    - 21.1.4 Single-user only tracing .................................... 563
    - 21.1.5 Consolidated log files and formats .......................... 563

#### 21.2 Pinpointing the problem .................................... 564
    - 21.2.1 Library Server .............................................. 565
    - 21.2.2 Resource Manager ........................................... 570
    - 21.2.3 Problems starting WebSphere Application Server on AIX 5L . 574
    - 21.2.4 WebSphere global security ................................... 578
    - 21.2.5 Resource Manager logging ................................... 579
    - 21.2.6 Secured Sockets Layer (SSL) ................................. 583
    - 21.2.7 Generating the Web server plug-in with SSL information for WebSphere Application Server ............................. 584
    - 21.2.8 Clients ....................................................... 587
    - 21.2.9 Installation .................................................. 590

#### 21.3 Troubleshooting TSM integration ............................ 593

#### 21.4 Traces on z/OS ................................................. 595
    - 21.4.1 Library Server log ......................................... 596
    - 21.4.2 Resource Manager trace .................................... 597
    - 21.4.3 Workstation log ............................................. 597

### Part 6. Appendixes .................................................. 599

#### Appendix A. Installation and configuration overview (for V8.2 only) .... 601
    - A.1 Installation overview for Windows (for V8.2 only) .............. 602
        - A.1.1 Content Manager prerequisites for Windows ................... 602
        - A.1.2 Set up users .................................................. 604
        - A.1.3 Configure Secure Sockets Layer (SSL) for IBM HTTP Server .... 605
        - A.1.4 WebSphere SSL configuration and localhost restriction ........... 606
        - A.1.5 Installing Content Manager on Windows ....................... 608
        - A.1.6 Installation verification ..................................... 611
    - A.2 Installation overview for AIX (for V8.2 only) .................. 612
        - A.2.1 Content Manager prerequisites for AIX ....................... 612
Notices

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

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

- AIX®
- AIX 5L™
- AS/400®
- CICS®
- ClearCase®
- DataJoiner®
- Domino®
- Domino.Doc®
- DB2®
- DB2 Connect™
- DB2 Universal Database™
- ImagePlus®
- IBM®
- Informix®
- Lotus®
- Lotus Discovery Server™
- Lotus Notes®
- Lotus Workflow™
- MVS™
- Notes®
- OS/2®
- OS/390®
- QBIC®
- Redbooks™
- Redbooks (logo) ™
- SecureWay®
- Tivoli®
- VideoCharger™
- VisualAge®
- VisualInfo™
- WebSphere®
- Word Pro®
- Workplace™
- Workplace Web Content Management™
- z/OS®

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

- ClearCase®, Rational®

The following terms are trademarks of other companies:


- EJB, Forte, Java, JavaBeans, JavaServer, JavaServer Pages, JDBC, JSP, JVM, J2EE, Solaris, Sun, Sun Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

This IBM® Redbook deals with IBM DB2® Content Manager Version 8.3 implementation and migration. It is aimed at architects, designers, developers, and system administrators of Content Manager systems.

In Part 1, we introduce Content Manager by providing the Content Manager products overview and architecture overview.

In Part 2, to help you better understand Content Manager, we cover the basic concepts needed to design and implement a Content Manager solution. This includes topics on data modeling, workflow, text indexing and search, application development, query language, security, and Tivoli® Storage Manager (TSM) overviews.

In Part 3, we cover the Content Manager solution implementation process from planning and designing, to deployment. To put concepts into real practice, we provide a practical case study to demonstrate how to implement a Content Manager solution for a real-world scenario.

In Part 4, we discuss Content Manager migration. This includes migration on multiplatforms, for TSM, and for Content Manager custom applications. In addition, we describe the approach and process for special migration scenarios such as cross platform migration and migration from a third-party product.

In Part 5, we discuss maintenance activities. Once a Content Manager system is implemented or migrated, it is important to maintain the system. We cover maintenance issues, including regular maintenance procedures, performance tuning, and troubleshooting hints and tips for a production Content Manager system.

By using this redbook, we hope you learn the basics that you will need to implement or migrate a Content Manager system. It is not our intention to cover all the Content Manager topics in detail. In many areas, we refer to the existing Content Manager product publications for reference. Please read the product publications in conjunction with this redbook.

Enjoy the world of Content Manager!
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, San Jose Center.

Wei-Dong Zhu (Jackie) is a Content Management Project Leader with the International Technical Support Organization at the Almaden Research Center in San Jose, California. She has more than ten years of software development experience in accounting, image workflow processing, and digital media distribution. She holds a master’s degree in Computer Science from the University of Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM DB2 Content Manager.

Jorge A. Andres is an Advisory IT Specialist for Information Management Group in IBM Mexico. He has over ten years of experience in the systems field, including programming and data management. He holds a degree in Computer Systems Engineering from UNAM University. His areas of expertise include applications development, Informix®, DB2, Content Manager, and Content Manager OnDemand. He is an IBM Certified Specialist and Solutions Expert for DB2 version 7.1, IBM Certified Database Associate and Database Administrator for DB2 Version 8.1, and IBM Certified Solution Designer for DB2 Content Manager Version 8.3. He has worked for IBM for six years.

Kenneth S. Christensen is a Senior IT Specialist for Software Group IBM Denmark. He has more than five years of experience in the Content Management area working in the Nordic Information Management Services team. He specializes in implementation, troubleshooting and teaching of Content Manager V8. He holds a Master of Science in Engineering from the Technical University of Denmark, he is an IBM certified solution designer for DB2 Content Manager V8, and he is certified in Domino®.Doc® System Administration 3.0. Kenneth has worked at IBM since 1994.

Chun Liu is an Advisory IT Specialist for Information Management Group in IBM China. He has more than five years of experience in the Content Management field. He holds a degree in Computer Science from Beijing University of Technology. He specializes in implementation, troubleshooting of Content Manager V8 on the Windows®, Linux, and AIX® platforms. He is an IBM Certified Solution Designer for Content Manager V8, a Certified Solution Expert of Content Manager OnDemand for Multiplatforms, a Certified Database Advanced Administrator for DB2 Version 8.1, and he is Certified for IBM DB2 Problem Determination Mastery. He has worked for IBM more than six years.

Glenn Dreves is a Certified Consulting IT Specialist for IBM Australia with more than seven years of experience in the Content Management field. He holds a degree in Business Information Technology from the University of New South
Wales, Australia, and is currently studying for his MBA at the Australian Graduate School of Management. Glenn specializes in Content Manager implementation on the Windows, Linux, and AIX platforms, and on application development using the Content Manager Version 8 APIs.

Special thanks to the following people who contributed to the first version of this IBM Redbook:

David Bartlett
Gerhard Fichtinger
Grant Myburgh
Gopal Sreeraman

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

Deanna Polm
Emma Jacobs
International Technical Support Organization, San Jose Center

Chandra Shekar
Mimi Vo
Don Benson
Anju Bansal
Badal Choudhary
Teresa Dain
Mahesh Garg
Nancy H Hang
Madhumati Krishnan
Therese McQuillan
Ken Nelson
Nicholas K Puz
Randy Richardt
Dave Royer
Phillip Sanchez
Sharon M Sanders
Sandi Shi
Tracee Tao
Parag Tijare
Phong K Truong
Celia Lin Tsao
Ganesh Vaideeswaran
Ali A Wasti
Alan Yaung
IBM Software Group, San Jose, CA
Become a published author

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

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online Contact us review redbook form found at:
  ibm.com/redbooks

- Send your comments in an Internet note to:
  redbook@us.ibm.com

- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099
Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7051-01
for Content Manager Implementation and Migration Cookbook
as created or updated on May 4, 2006.

April 2006, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information in IBM DB2 Content Manager for Multiplatforms Version 8.3 as described below.

New information
- XML support
- Import/Export utility

Changed information
- Product overview
- Data modeling
- Workflow
- Text indexing and searching
- Query Language
- Security
- Upgrade and migration on multiplatforms
- Special migration scenario
- Application migration
- Maintenance
- Performance tuning
- Troubleshooting
- Configuration and log files
Unchanged information
The previous edition of this IBM Redbook reflects IBM DB2 Content Manager Version 8.2. In this edition, all chapters have been updated to reflect IBM DB2 Content Manager Version 8.3, except the following chapters:

- TSM overview
- TSM migration
- Migration on z/OS®
- Installation and configuration

Since the publication of the first edition of this IBM Redbook, there is a new redbook specifically for Content Manager on z/OS. To avoid duplication of the materials, the migration on the z/OS chapter in this redbook is not updated. To get information on this topic, refer to this new redbook:

- *Content Manager for z/OS V8.3 Installation, Implementation, and Migration Guide*, SG24-6476

Content Manager Version 8.3 has achieved great improvements in product installation and configuration. It is much easier to install the Content Manager Version 8.3 product than for Version 8.2. We do not feel it is necessary to duplicate what is already well documented in the online Information Center and the product manual. To get the latest information on installation and configuration, refer to:

- IBM DB2 Content Manager V8.3 Information Center:
  
  [http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0](http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0)

- *IBM DB2 Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332

The out-dated chapters (Migration on z/OS, Installation, and Configuration) have been moved to the appendix section. They are still kept in this redbook for reference purposes, in case you might need to set up a Content Manager Version 8.2 environment.
Introduction

In this part of the book, we introduce Content Manager by providing an overview of Content Manager products as well as an architecture overview.
Content Management products overview

Over the last decade, there has been widespread adoption of e-business solutions. Companies need to manage information, processes, knowledge, and business operations electronically; the systems that manage these areas have been developed over time.

These systems have been known variously as content management, document management, knowledge management, collaboration management, digital asset management, and digital rights management. IBM has been a leader in providing solutions for several of these areas. In this chapter, we discuss the complete and powerful Content Manager product portfolio that IBM delivers.
1.1 Content management portfolio

During the last 15 years, IBM has developed and implemented several Content Manager solutions across thousands of customers. A solution may address a unique problem or business requirement and hence may vary from other solutions. More often than not, a combination of these products is utilized to implement a customized content management solution. Here are some key products and solutions that IBM has in its content management delivery portfolio:

- IBM DB2 Content Manager
- IBM DB2 Content Manager OnDemand
- IBM DB2 Content Manager VideoCharger™
- IBM DB2 Content Manager CommonStore
- IBM DB2 Document Manager
- IBM DB2 Records Manager
- IBM WebSphere® Information Integrator for Content Edition
- IBM Workplace™ Web Content Management

1.1.1 IBM DB2 Content Manager

The IBM Content Management portfolio is a suite of products and offerings designed to manage volumes of electronically captured content that support mission critical business processes.

The heart of the portfolio is IBM DB2 Content Manager. With a proven history as a highly-scalable, production strength document imaging system designed to capture, process, and store vast amounts of scanned paper-based information, Content Manager has evolved into a repository for virtually any type of digital content, including forms, scanned images, electronic office documents, HTML-based and XML-based Web content, high volume e-mail archives, and rich media such as digital video and audio.

Unlike simple file systems, Content Manager uses a powerful relational database, DB2 or Oracle, to provide indexed search, security, and lifecycle management services. Content Manager provides check-in and check-out, version control, object-level access control, a flexible data model that enables compound document management, and advanced searching based on user defined attributes. It also includes workflow functionality, automatically routing and tracking content through a business process according to predefined rules.
A Content Manager implementation is typically composed of one Library Server and one or more Resource Managers. The Library Server responds to user queries, while the Resource Managers maintain collections of content. Deploying remote Resource Managers can keep content close to its point of use, reducing bandwidth requirements and increasing disaster protection, yet still keeping content under central management control — crucial for risk and compliance reasons.

Content Manager is integrated with and ready for use with leading scanning sub-systems and other capture solutions requiring high throughput of large volumes of data. This tight integration and the scalability inherent in the Content Manager architecture allow implementation of systems that manage very high volumes of data for purposes such as e-mail archiving. Furthermore, Content Manager’s open and documented APIs (C++ and Java) and Web services interfaces provide a developer with multiple options when building a custom application.

**IBM DB2 Content Manager Information Integrator for Content**

IBM DB2 Content Manager Information Integrator for Content (formerly known as Enterprise Information Portal) is a *component* that comes with IBM DB2 Content Manager. It integrates and gives access to multiple repositories spread across a network. This includes unstructured content such as XML, HTML, and streaming video, and structured data information present in enterprise relational databases. Information Integrator for Content uses the Lotus® Extended Search to give access to Lotus Notes® Domino databases and various Web search engines. It also can access the business objects within an IBM WebSphere MQ Workflow process.

Information Integrator for Content enables knowledge workers to concurrently access data and content from multiple sources from a Web browser. Information Integrator for Content provides a federated search capability to enable transparent searches across multiple systems such as Content Manager, OnDemand, Lotus Domino.Doc, DB2, other relational databases, FileNet Panagon imaging services, and data warehousing sources. Information Integrator for Content provides the Lotus Extended Search for scalable and distributed searches across Domino, Microsoft® Exchange, search indexes, LDAP sources, file systems, and Web-based content.
Information Integrator for Content also provides advanced Web crawling and Information Mining features. The Web Crawler, which is part of the Information Mining component of Information Integrator for Content, can crawl over 200 types of content on the Internet, Intranet, extranet, file systems, Lotus Notes, Domino, and FTP collections. The collection of the crawled content can be processed by the Information Mining component for further analysis. This tool provides many functions, including tools for categorizing and generating enterprise taxonomies, document summarization, language identification, information extraction, clustering like content, and providing advanced search on the imported collection.

1.1.2 IBM DB2 Content Manager OnDemand

IBM DB2 Content Manager OnDemand is a high performance repository optimized for managing computer output. OnDemand provides a highly reliable and flexible system to meet data archive and retrieval requirements. It can store and index about 2 million pages per hour, the performance demanded by high volume billing or statement processing applications. OnDemand transforms any type of print output format, such as invoices, customer statements, bills, reports, and check images, into searchable, Web-integrated, electronic content that can be deployed in a variety of ways to meet customers requirement and resolve their problems. OnDemand allows computer print output to be bundled, redirected over the network, and automatically distributed based on business rules.

One of the key strengths of OnDemand is its ability to directly archive computer printout data streams. It can then display the invoices and reports to users, and do statement presentment on the Web or portal. Access control is down to the granular level. Desktop and Web plug-ins are available to provide multiple conversion approaches for flexible viewing and delivery options. Interfaces to high volume print engines are also provided. OnDemand is optimized to capture, search, present, and manage large collections of small objects such as statements or bills.

Document capture is fast and unattended with multiple data types such as line data, Xerox meta code, and optionally PDFs. Index values are automatically extracted and stored in the database. Application report templates are predefined for easier administration. Access to the documents is provided by means of convenient numerous search and presentation options.
Since storing large amounts of documents requires space, OnDemand works tightly with the Tivoli Storage Manager (TSM) to offer the right solution of mixed needs of immediate access, document retention requirements, and lower cost of storage. Built-in security lets administrators control access to the system or access to reports and documents, which can be limited by type or section of a report. For example, a user can be restricted to statements for specific account numbers, and individual functions, such as printing, can also be restricted.

OnDemand has been used in many solutions such as Electronic Statement Presentment (eSP), Electronic Bill Presentment and Payment (EBPP) and Customer Service for Call Centers.

1.1.3 IBM DB2 Content Manager VideoCharger

IBM DB2 Content Manager VideoCharger, which is currently delivered as a feature of the IBM DB2 Content Manager product, provides for real-time multimedia streaming. It supports multimedia formats including MPEG-4 and supports IP multicast and real-time encoding. VideoCharger can be used to provide a high quality streaming and broadcasting environment for intranet as well as Internet users. The VideoCharger pushes video to the client over the network similar to a broadcasting network. This compares favorably with some file servers where data is pulled by the client in successive reads.

VideoCharger also has a player that can be downloaded at the client to control the playback of video and audio streams received from the VideoCharger server. VideoCharger supports a wide range of media formats such as AVI, WAV, MPEG-1, and MPEG-4, and offers high scalability on many platforms, including AIX and Windows. The accompanying tools enable monitoring for performance and enable gathering auditing metrics on access and hits of media objects.

VideoCharger, integrated with Content Manager, provides for a unified digital media management system for all media assets.

1.1.4 IBM DB2 Content Manager CommonStore

IBM DB2 Content Manager CommonStore helps to seamlessly integrate SAP, Lotus Domino, and Exchange Server with IBM archives. CommonStore integrates with the target system to off-load data on to an external storage. This improves the performance of the target system and cuts down storage costs.

Three components available for CommonStore are:

- CommonStore for Exchange Server
- CommonStore for Lotus Domino
- CommonStore for SAP
CommonStore for Exchange Server helps with E-mail archival and retrieval. It manages e-mail server growth by automating e-mail archival, thus trimming down the size of online E-mail storage. This is accomplished by Windows services that are available to automate tasks. Archival can be configured to archive the entire mail document including attachments, or only the attachments. Also provided is the facility for end users to directly access archived E-mails using a Web browser or Outlook® client. The E-mail documents are accessible using the Content Manager Client for Windows or the OnDemand client.

CommonStore for Lotus Domino helps with the same productivity enhancements as its Exchange Server counterpart described above. Apart from helping automated archives of documents, attachments, views, and folders, these archives can also be made available for text search using the Lotus Discovery Server™. Similarly, the archives are accessible using the Lotus Notes client, Content Manager client, or a Web browser.

As an SAP-certified solution, CommonStore for SAP off-loads data, trims the size of the database, and improves system performance. It supports all SAP operations databases such as DB2, Informix, and Oracle. It manages all types of data and documents defined in the SAP ArchiveLink. Integration is provided with SAP applications such as Workflow, Document Management System (DMS) and SAP R/3 Document Finder.

### 1.1.5 IBM DB2 Document Manager

IBM DB2 Document Manager (Document Manager) addresses the origination, approval, distribution, and revision of unstructured information. In the simplest terms, document management enables electronic files to be reused or repurposed by multiple individuals.

Document Manager offers a Web-based Document Manager Desktop client and a set of the tools to provide enterprise-scale document management services coupled with a robust, scalable, enterprise-capable repository such as Content Manager or Lotus Domino.Doc.

The key capabilities of DB2 Document Manager are:

- Rules-based document life cycle management
- Desktop application integration
- Compound document support
- Engineering file formats support
- Renditioning services
- Remote printing/plotting
- Process management
- Bulk document loading
- Automated notification
Document Manager manages both simple and compound documents. Compound documents are documents composed of multiple components such as a Microsoft Word document with an embedded Microsoft Excel® spreadsheet, and an embedded GIF image. Each component's life cycle and security is managed by Document Manager individually while the appropriate interrelationships are maintained. These interrelationships of managed components may affect the retention of individual components.

Through its integration with Records Manager, Document Manager ensures that the appropriate business rules are applied when related documents of record are processed. Each component is efficiently stored and managed in the Content Manager repository while exploiting the data modeling capabilities of the repository to create the interrelationships and maintain referential integrity of these related components.

### 1.1.6 IBM DB2 Records Manager

IBM DB2 Records Manager (Records Manager) is an application and also an embedded engine that enables the management of enterprise’s electronic and physical records throughout their records life cycles. Records Manager delivers the record keeping functions through the embedded engine technology. It is integrated with applications such as IBM DB2 Content Manager, IBM DB2 Document Manager, IBM DB2 CommonStore for Lotus Domino, and IBM DB2 CommonStore for Exchange Server. Records Manager APIs facilitate the integration with any application that requires its record keeping capabilities.

With Records Manager, a business can records enable virtually any application, from commercial to custom-built. All the underlying record keeping infrastructure and processes are supplied by Records Manager.

Records Manager provides the following record keeping functions:

- Corporate records declaration and classification
- Records life cycle management
- Record metadata management
- Record content searching, retrieval, and viewing (including text search)
- Document auditing and reporting
- Users and security management

Some of the reasons why you want to integrate Records Manager with your business applications include:

- Meeting compliance requirements
- Supporting required business processes, procedures, and standards
- Embedded engine technology, seamlessly integrated with other applications
- Web based administration client
► API integration using Java™, C++, and .Net
► Scalable architecture
► Flexible file plan design
► Content maintained in the host application’s repository
► DoD 5015.2 and PRO certified

With Records Manager, you can declare and classify records from using fully automatic procedures to manual processing.

1.1.7 IBM WebSphere Information Integrator Content Edition

IBM WebSphere Information Integrator (II) Content Edition delivers federated access within portals, workflow, and other enterprise applications, to information stored in disparate content management systems from IBM and other vendors such as FileNet, EMC/Documentum, and Open Text, from within portals, workflow, and other enterprise applications. Information Integrator Content Edition makes multiple disparate repositories look and act as a single unified repository, and provides a complete platform for deploying applications and workflows spanning multiple content sources.

WebSphere Information Integrator (II) Content Edition includes:

► Federated access to multiple disparate content management and workflow systems
► Complete platform for deploying repository-spanning applications and workflows
► Bidirectional access to content and workflow, as well as the underlying functionality
► Pre-built integrations to more than 20 content management and workflow systems
► A rich set of functions spanning multiple repositories, including federated search
► Being built on a service oriented architecture (SOA) and standards based

WebSphere Information Integrator Content Edition comes with out-of-the box connectors and toolkit.

The out-of-the box connectors are available to leading content repositories to quickly unify a broad range of content sources and workflow systems without the cost, complexity, and risk of custom programming efforts.
Connectors are available to the following systems:

- IBM DB2 Content Manager
- IBM DB2 Content Manager OnDemand (both distributed and z/OS)
- IBM WebSphere MQ Workflow
- IBM WebSphere Portal Document Manager (read-only)
- IBM Lotus Notes
- Documentum Content Server
- FileNet Content Services
- FileNet Image Services
- FileNet Image Services Resource Adapter
- FileNet P8 Content Manager
- FileNet P8 Business Process Manager
- Open Text Livelink
- Microsoft Index Server/NTFS
- Stellent Content Server
- Interwoven TeamSite
- Hummingbird Enterprise DM
- Read-only access to the following relational database systems:
  - DB2 Universal Database™
  - Oracle
  - Any database accessible through WebSphere Information Integrator federated data server

WebSphere Information Integrator Content Edition toolkit lets users develop, configure, and deploy content connectors to additional commercial and proprietary repositories.

1.1.8 IBM Workplace Web Content Management

IBM Workplace Web Content Management™ delivers end-to-end Web content management for Internet, intranet, extranet and portal sites. By leveraging content in back-end systems, Workplace Web Content Management reduces development and implementation time and places content creation and management firmly in the hands of content experts for “author once, publish everywhere” control. The product runs on both Lotus Domino and IBM WebSphere and provides for the integration of IBM WebSphere Portal and IBM DB2 Content Manager.

Workplace Web Content Management is a powerful tool that helps you generate, store, and serve Web content. Here are some of the benefits it offers:

- Streamlines the Web content management process from content authoring, workflow, management, integration and delivery
- Publishes information on demand in minutes, not days, for improved responsiveness to customers, partners, suppliers, and employees
1.2 Choosing the right solution

Your IBM representative should assist you in selecting the right product and choosing a suitable solution set for your business requirements. As a quick reference, we describe some characteristics of the core IBM content management products that may help to meet your unique business requirements.

1.2.1 Content Manager

Content Manager offers the following benefits for your business needs:

- Scalable repository services, designed for high volume applications
- Sophisticated data model
- Capability to store and update content. Supports check in, check out, and versioning
- Integration with IBM Records Manager for compliance solutions
- Support for centralized or distributed architectures
- Out of the box integration with leading capture sub-systems
- Support for streaming video and audio applications
- Built-in hierarchical storage management
- Choice of clients include Windows, thin (browser), portlet or create a custom client with the CM API
- Document routing capability with documented APIs
- Java and C++ APIs and Web services interface; Visual & non-Visual Beans
- Convenient capture sub-systems such as Kofax Ascent Capture
1.2.2 OnDemand

OnDemand offers the following benefits for your business needs:

- High volume imaging and very high speed ingest rates with multiple check-printing and computer output data streams
- Convenient capture sub-systems such as Kofax Ascent Capture
- No streaming video and audio capability
- Desktop and Web clients for viewing
- TSM integration for device attachment
- High integration with Electronic Bill Presentment and Payment (EBPP)

OnDemand is best for computer output capture, print data stream capturing and viewing, check imaging, and statement presentment system.

OnDemand is appropriate for applications that do not need the capability to update content or document management functions such as check in and check out. Content stored in OnDemand is typically for archival purposes.

1.2.3 Document Manager

Document Manager offers the following benefits for your business needs:

- Tight integration with Microsoft applications and email applications.
- Sophisticated handling of documents generated with PC applications, including managing access and security and approval processing.
- Capability to handle multiple versions and revisions of the same document, manage check-in and out, and update capability.
- Need the ability to render into another format, for example a non-editable format, for review purposes.
- Want to automate document processing through the use of document and processing templates.
- Want to include desktop applications in a risk and compliance solution (records management)

1.2.4 CommonStore

CommonStore for SAP offers the following features for your business needs:

- Ensure high performance of SAP databases.
- Alleviate growth of the SAP datastore by archiving old, closed transactions.
- Keep quality assurance and test more similar by keeping less unneeded data in production.
- Integration of content to support business processes in SAP workflow (incoming/outgoing documents).

CommonStore for Exchange/Domino offers the following features for your business needs:
- Ensure top performance by removing old data from mail servers.
- Reduce hardware purchases just to keep up with the growing volume of mail.
- Provide more efficient backup and recovery of mail systems.
- Provide a Records management capability for email systems.

1.2.5 WebSphere Information Integrator for Content Edition

WebSphere Information Integrator for Content Edition offers the following benefits for your business needs:
- Provide integrated access to multiple data repositories.
- Provide the ability to add federation services such as metadata mapping, federated search, cross-repository event management and single sign-on to data stores.
- Deliver content to external business value networks, such as customers, suppliers and manufacturers.
- Future-proof applications by ensuring they are independent of underlying content sources, and enables repositories to be added, modified or removed without disrupting end applications.
- Speed integration of new data stores, after, for example, a merger or acquisition.

1.2.6 Workplace Web Content Manager

Workplace Web Content Manager offers the following benefits for your business needs:
- Rapidly develop and manage intranet, extranet, Internet and portal assets.
- Unleash enterprise content into WebSphere Portal Server by rapid portlet development without extensive programming.
- Optimize and integrate Web applications with existing back-end systems.
1.3 What’s changed in Content Manager V8.3

There are a number of changes in Content Manager V8.3, which continues to deliver a real return on investment to customers. Version 8.3 focuses on five areas: integration, open systems, autonomic systems, resiliency, and ease of use. These highlights, and other enhancements to the Version 8.3 product, are summarized in the following sections.

**Support for Oracle databases**
Content Manager V8.3 adds support for Oracle databases managing the metadata stored in both Library Server and Resource Manager. Migration tools are included for Oracle users of Content Manager V7.

**Remote database server for DB2 UDB and Oracle**
You can now help reduce workload by installing the Content Manager Resource Manager database on a different machine than the Resource Manager application.

**System administration client for AIX and Sun Solaris**
Content Manager V8.3 adds support for the system administration client to run on the AIX and Sun™ Solaris™ operating systems.

**Web services support**
Content Manager provides a self-contained, self-describing modular interface, a Web services interface, that you can use within your applications, with other Web services, or in complex business processes to seamlessly access items stored in Content Manager. The Web services interface allows you to integrate dynamically your applications with Content Manager, regardless of the programming language they were written in and the platform they reside in.

**XML support**
Content Manager includes enhanced support for XML data.

The API includes XML-specific calls that allow Content Manager to quickly and efficiently process XML data streams. It also includes a sophisticated mapping tool that makes it easy to define the Content Manager data structures required to store an incoming XML-data stream. The Content Manager structure can be automatically generated from an XML schema from the other application, greatly reducing the time and effort required for configuration.
The system administration client enables you to import and export system configuration data from or into an XML file. This capability allows you to easily copy administrative settings from one server to another by exporting the information and importing it into the other systems. You can also use the new XML capability to get the list of system administration objects such as users or groups from one Content Manager system to another.

**Document routing enhancements**

Content Manager document routing is enhanced in V8.3 to include decision points, actions, action lists, parallel routing, and user exit support. In addition, a new graphical builder within the system administration client helps you easily define your document routing processes.

**Query (search) enhancements**

The query function is enhanced to include the following support:

- Query on checked-out items
- Row-based view filtering in query
- Get the count of query results without getting the results themselves
- Use of the IN operator to compare an attribute’s value to a list of values
- Internal query optimization to reduce the length of the generated SQL statements

**Consistent logging and tracing interface**

V8.3 provides consistent logging and tracing interface that covers most system components:

- The system administration client now provides the log control utility, which you can use to set log and trace parameters for multiple system components
- A default common directory for all log files
- A standard log file timestamp format using Greenwich Mean Time
- Support for logging information related to a single user ID
- A unique log ID that is common among different system component log files

**Installation improvements**

The following installation improvements are now available:

- Automated user ID creation. If selected, default administrative users will be created locally and added to the appropriate groups.
The installation programs for Content Manager, including the Information Integrator for Content and eClient components are redesigned to provide commonality for all operating systems, consistent product interoperability, and an improved, more robust, installation experience.

You can selectively install features of the products and some features are sharable among the products.

The installation programs for Content Manager now include time saving typical installation paths, which greatly reduce the complexity of user input for common installations. The custom paths are reorganized to improve clarity and consistency.

Silent installation capability is consistently supported for all products and operating systems, allowing for the full range of installation options.

Pre-requisite checking is redesigned to be more flexible and precise, thus facilitating a wider range of installation topologies, and allowing the flexibility to extend capabilities.

Automatically configures the Secure Sockets Layer (SSL) of the IBM HTTP Server shipped with WebSphere Application Server Version 5.1 and used by the Content Manager Resource Manager.

Online help is available for installation panels that provides information about default values and limitations for fields, and relevant background information.

**Elimination of C++ compiler dependency**

C++ compiler is no longer required to install Content Manager. All item creations, reads, updates, and deletions are performed with dynamic SQL.

**Discontinued and deprecated function**

The following functions are no longer supported in Information Integrator for Content in V8.3:

- Extended Search
- IBM Content Connector for Panagon Image Services
- IBM Web Crawler
- Connectors for:
  - Information Catalog Manager
  - Extended Search
  - DataJoiner®
- ActiveX® versions of the following connectors: Content Manager V7 (DL), Lotus Domino Doc (DD), OnDemand (OD), Extended Search (DES), VisualInfo/400 (V4), Image Plus/390(IP) and Federated (Fed))
Accessibility improvements
Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. Version 8.3 enhances product accessibility features. For example, new short cuts have been added to help you operate all features using the keyboard instead of the mouse.

Object storage
In Content Manager V8.3, two new device managers are introduced, ICMBLOB and ICMCIFS. These device managers are used to store objects directly into a table in the database or to a Network Attached Storage.

- ICMBLOB is a device manager used to store the object into the database as a BLOB data type.
- IBMCIFS is a device manager used to store objects on the NAS device.

Filtering
You can use filtering for viewing users, user groups, access control lists, and item types in the system administration client.
Content Manager architecture overview

In this chapter, we provide a high level overview of IBM DB2 Content Manager Version 8 and explain how each component works with the others.
2.1 Architecture

The IBM DB2 Content Manager solution has been designed for Enterprise Content Management (ECM) objectives such as:

- Out-of-the-box ECM functions such as versioning, workflow, and security
- Rich functionality, such as multiple content formats and streaming video and audio capability
- Flexible content data model, taxonomy, and associated security
- Object-oriented Application Programming Interfaces (API) to allow extensions and integration between Content Manager and line-of-business systems
- Scalable architecture with small memory footprint to support high performance
- Lower TCO with tools for easier deployment and system administration
- Robust and reliable architecture to fit in with infrastructure tasks such as backup, replication, failover, load balancing, and troubleshooting

To achieve these objectives, Content Manager utilizes a patented architecture known as a triangular architecture, as shown in Figure 2-1.

![Figure 2-1 Content Manager triangular architecture]

The core component of Content Manager is the Library Server and Resource Managers. Applications (thick or thin client) use object-oriented APIs to invoke all Content Manager services that are divided between one Library Server and one or more Resource Managers.
When content is created and stored, it is physically stored on a Resource Manager. Content metadata and access control is managed by the Library Server using its own DB2 or Oracle database repository and store procedures. Clients store or retrieve content via Library Server. When a client requests to retrieve an object, the Library Server performs the query against its own database, then passes the result back to the client with the object token and resource location for which the user is authorized. The client will retrieve through the API, or it can communicate directly with the Resource Manager to retrieve the object, using any of the standard protocols such as FTP, HTTP, or FILE.

### 2.2 Content Manager components

As depicted by the triangular architecture, Content Manager consists of the Library Server, Resource Managers, and clients accessing Content Manager functionality. In this section, we briefly cover each main component.

#### 2.2.1 Library Server

The primary job of the Library Server is to service client requests for content. It manages content meta data and access control in a DB2 or Oracle database. DB2 is a relational database management system (RDBMS) that comes embedded with Content Manager. The Library Server utilizes many of the robust features of DB2. In many cases, the Library Server code, the code that typically run in the business logic tier, is implemented as stored procedures. The database is the heart of the Library Server and is always installed on the same machine as the Library Server application.

The database is also the key to designing and implementing the basic content data model. Content Manager, in association with DB2, provides a flexible data model that can support enterprise taxonomies. This data model allows a Content Manager solution to have powerful building blocks for items and objects. Items are defined by item types and have individual attributes describing them. For example, an Insurance Claim item type has attributes such as Claim Number, Date, and Customer ID. Attributes can reference each other, much like the foreign keys in a standard RDBMS.

The Content Manager architecture provides these data model features:

- Flexible data model to define item types and attributes as required by business processes
- Hierarchical, parent and child, and peer-to-peer relationships to express real-world environments
- Links and references
Links can model many-to-many relationships, and a link can be traversed bi-directionally. For example, a customer and an insurance claim can be linked by a “has” relationship. The relationship can be defined as a link. References are Content Manager data types that set referential integrity between attributes. For example, a customer ID in a CRM database table can be used to validate a customer ID entered in an insurance claim item in Content Manager. References also help data integrity by preventing deletion of a piece of information when something else is depending on it.

Content Manager has a versioning feature. Since each version of a document is stored as a separate item, it is easy to retrieve the most recent or a specific version. The system administrator can limit how many versions can exist for an item. When the limit is exceeded, old versions are automatically deleted.

For detailed information on data modeling, refer to Chapter 3, “Data modeling” on page 29.

Content Manager provides powerful search and access technologies. It has three types of searches that can quickly help users to locate content:

- **Parametric search**: Searching content based on metadata attributes such as account number and vendor ID.

- **Free text search**: Searching with free text or keywords to locate documents that contain the search term anywhere in its body.

- **Combined search**: Combination of parametric and free text search.

If an item type is defined as full-text searchable, then an item of that type is automatically indexed. This is made easier by DB2’s inherent administration facilities. Apart from providing document text search capability, Content Manager also allows free-text or partial keyword searches against metadata attribute values. This powerful search capability is provided by leveraging the DB2 Net Search Extender (NSE) component. The NSE automatically creates an index for each attribute and item type that has been defined as text-searchable. Whenever the Library Server receives a parametric or free text or combination query, all it has to do is pass the query to the DB2 database engine.

For detailed information on search, refer to Chapter 5, “Text indexing and searching” on page 117.

As described in the triangular architecture diagram, clients connect to the Library Server and use APIs to issue SQL queries. The APIs also incorporate a query language based on the XQuery path expressions, which is based on XMLPath expressions. This query language helps to easily navigate through the hierarchical data models.
For detailed information on the query language, refer to Chapter 7, “Query language” on page 161.

### 2.2.2 Resource Manager

A Resource Manager is the repository for content managed within Content Manager. There can be more than one Resource Manager to provide for additional repository for content, for failover, and for availability. A Library Server can connect to multiple local or remote Resource Managers. Client requests always go through the Library Servers first which ensures access control to objects. After authentication and authorization, the client accesses content from the Resource Manager. Both the Library Server and the Resource Manager can be configured to authenticate with the same LDAP directory for single sign-on purposes. A Resource Manager is basically a WebSphere Application. The client communicates with the Resource Manager using standard HTTP, FTP, and FILE protocols.

Similar to the Library Server, the Resource Manager also utilizes a DB2 database to manage the location of storage objects, locations, and devices. The Resource Manager works closely with TSM to define storage classes and migration policies. For example, once an object has been stored on a magnetic disk, it can be automatically migrated to a tape disk, for example, after six months, to reduce enterprise storage costs. The migrator facility helps the migration of objects from one defined storage class to the next. This migration capability is useful when moving a system from pilot to production environment, and scaling to a server with higher capacity as business growth demands.

For more information on TSM, refer to Chapter 9, “Tivoli Storage Manager for Content Manager” on page 235.

There is another function of the Resource Manager, *Replication*, that can selectively replicate document collections from one Resource Manager to another. For example, you can designate one Resource Manager as the primary and designate another one as the secondary. The two can be synchronized by using the replication feature for the document collections.

Resource Managers can be deployed in a geographically distributed manner so that frequently accessed objects can be retrieved faster. The Library Server will direct the client or API to retrieve objects from the Resource Manager. Whenever an object is not found on the local server, the Library Server retrieves the object from the appropriate remote Resource Manager and stores the object in the staging directory of the local Resource Manager servers. This is an optional feature and is known as a *LAN cache*. 
In addition to managing documents and images, a Resource Manager can manage streaming video and audio objects by integrating with the IBM Content Manager VideoCharger.

With Content Manager comes Information Integrator for Content, providing a wide range of connectors to access content in other repositories outside of the Resource Manager. Examples are relational database, Content Manager Version 7, DB2 Content Manager OnDemand, ImagePlus® for OS/390®, and Domino.Doc. You can use each native connector to access the repositories directly or you can use the federated connector to provide a generic unified interface to multiple repositories. With this capability, clients or applications can search or access multiple repositories as if they were a single virtual content repository.

### 2.2.3 Content Manager clients

Content Manager provides out-of-the-box clients: System Administration Client, Windows client, and eClient. System Administration Client is used to manage and perform administrative tasks on a Content Manager system. Windows client provides end users with Content Manager functionalities. eClient is a Web browser-based client that provides similar functionalities as the Windows client.

Content Manager Portlets are also available to create portal clients. In addition, Content Manager provides APIs that you can use to create a customized client application.

**The System Administration Client**

The System Administration Client for Windows is a desktop, Java-based application that assists in configuration and system maintenance. It runs on Windows 98 or later desktops, on UNIX platforms, and is built using the Java version of the Content Manager object-oriented APIs. The System Administration Client allows you to manage Library Servers, Resource Managers, data models (such as item types and attributes), security (such as users, groups, access control lists, and privilege sets), storage management, and document routing (such as workflow).

User management can use a centralized LDAP or Windows integrated authentication for all Content Manager applications. The System Administration Client can configure Content Manager to use an LDAP server for all authentication. The users and groups in the LDAP can be imported into Content Manager by the System Administration Client or any batch utility. This provides for the granular object-level access control for authorization.
Content Manager also allows exit routines to be used for integrating with custom authentication mechanism for non-LDAP servers. There is also the ability to create administrative domains on the Library Server to manage groups of users. Administrative domains streamline and distribute user management in a Content Manager configuration with a large user base divided among many departments.

For detailed information on security, refer to Chapter 8, “Security” on page 187.

**Windows client**

Windows client, also known as Client for Windows, is a desktop client that provides out-of-the-box capabilities for supporting production-level Content Manager applications. Using the Windows client, you can import or export documents to the file system. While importing one or more items, you can specify the item type and associated attribute values. Content Manager automatically adds the item to the system and can automatically start a workflow process. Users can also start a workflow process manually by specifying the process name and associating a priority to it. The base name of the document is combined with other values by Content Manager to uniquely identify the object.

For detailed information on workflow, refer to Chapter 4, “Workflow” on page 75.

Content Manager supports the Open Document Management API (ODMA). You can use any ODMA-enabled application and use simple functions such as **File → Save** and **File → Open** to import and export document in Content Manager.

The client also provides rich document and image annotation options, including pen, highlighter, box, circle, arrow, and text notes.

**eClient**

eClient is a browser-based client that provides out-of-the-box capabilities for Content Manager systems, similar to that of the Windows client. Import and export can be done using the eClient wizards. Document and folder organization functions are available through eClient. The eClient viewer provides page navigation function such as next, prev, last, goto, zoom in, and zoom out.

The application displays the first page of a document as soon as it is available without waiting for the entire document to download. This improves the response time when working with large documents. eClient also supports the same type of search capabilities as a Windows client.
**Portal client**

Portal client is an extensive and customizable front end to Content Manager. It is composed of Content Manager Portlets, a Web-based application, running on the WebSphere Portal environment.

Content Manager Portlets consists of two portlets: the Main portlet and the Viewer portlet.

The **Main portlet** provides a single portlet-based functional replacement for the current Content Manager eClient. When deployed alone, the Main portlet will display documents in new browser windows for viewing.

The **Viewer portlet** can optionally be deployed on the same portal page as the Main portlet to provide document viewing capability with a unified layout in an integrated fashion, instead of in separate browser windows. Click-2-Action (C2A) is also supported by the Viewer portlet. Other portlets, including the Main portlet, can communicate to the Viewer portlet using C2A to have a document displayed in the Viewer portlet.

### 2.3 Architecture extensions

Content Manager provides powerful application development toolkits that include C++ and Java APIs. It also has a rapid application development toolkit that includes Java Beans (visual and non-visual) that can be utilized by Servlets and Java Server Pages (JSPs).

As discussed earlier, Content Manager provides a federated connector and native connectors to access a wide variety of content repositories. The exit routines provided within the document routing process can execute and integrate with many types of external line of business applications. Since Content Manager supports the W3C standard XPath/XQuery-based XML-document management, developers can easily build applications that can navigate and traverse the entire content model easily.

For more information on developing a customized Content Manager application, refer to Chapter 6, “Application development overview” on page 131.
Understanding the product

In this part of the book, we cover the basic concepts needed to design and implement a Content Manager solution. This includes topics on data modeling, workflow, text indexing and search, application development, query language, security, and Tivoli Storage Manager (TSM) overviews.
Data modeling

A thorough understanding of data modeling for Content Manager is required before designing and implementing a Content Manager solution. In this chapter, we introduce data modeling for Content Manager. We discuss the data modeling building blocks and demonstrate how to use the System Administration Client to implement them. In addition, we make comparisons with the Content Manager Version 7 data model and highlight the enhancements in Version 8.
3.1 Data modeling entities

Content Manager Version 8 supports multi-level, hierarchical data structures that accommodate complex meta data hierarchies. In addition, Content Manager provides the ability to model one-to-many relationships between content and its attributes, including multi-valued attributes. Custom applications can also be built to take advantage of complex inter-item relationships with full referential integrity.

In this section, we describe the various data model elements in Content Manager, such as attributes and items. Note that certain data model elements are not supported in the Windows Client or the eClient.

Table 3-1 lists the data model elements and their level of support in the clients (Windows Client 8.3 and eClient 8.3).

Table 3-1  Data modeling entities and client support level

<table>
<thead>
<tr>
<th>Data modeling entity:</th>
<th>Supported by:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows Client 8.3</td>
</tr>
<tr>
<td></td>
<td>eClient 8.3</td>
</tr>
<tr>
<td>Attribute</td>
<td>Yes^{a}</td>
</tr>
<tr>
<td></td>
<td>Yes^{b}</td>
</tr>
<tr>
<td>Attribute group</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Root component</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Child component</td>
<td>One level only</td>
</tr>
<tr>
<td></td>
<td>One level only</td>
</tr>
<tr>
<td>Media object class</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>MIME type</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Item type classification: item</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Item type classification: resource item</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>No^{c}</td>
</tr>
<tr>
<td>Item type classification: document</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Item type classification: document part</td>
<td>Yes^{d}</td>
</tr>
<tr>
<td></td>
<td>Yes^{e}</td>
</tr>
<tr>
<td>Item type subset f</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Links</td>
<td>Folders only</td>
</tr>
<tr>
<td></td>
<td>Folders only</td>
</tr>
<tr>
<td>Foreign keys</td>
<td>Yes^{g}</td>
</tr>
<tr>
<td></td>
<td>Yes^{h}</td>
</tr>
<tr>
<td>References</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Semantic type</td>
<td>Yes^{i}</td>
</tr>
<tr>
<td></td>
<td>Yes^{j}</td>
</tr>
<tr>
<td>Versions</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
3.1.1 Attributes

An attribute describes a characteristic or property of an item. The attribute can be searched on to locate that item. Table 3-2 lists some typical attributes to describe a student.

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Student Number</th>
<th>Date of Birth</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Doe</td>
<td>s1234576</td>
<td>12/12/1973</td>
</tr>
<tr>
<td>Mary</td>
<td>Smith</td>
<td>s2083571</td>
<td>01/07/1969</td>
</tr>
<tr>
<td>Robert</td>
<td>Washington</td>
<td>s9801223</td>
<td>03/12/1976</td>
</tr>
</tbody>
</table>

In this case, First Name, Last Name, Student Number, and Date of Birth are four attributes that describe a student.

To create an attribute, use the following procedure:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Attributes from the left-hand tree menu.
2. Select from the file menu, select Selected → New. Alternatively, right-click Attributes and select New.
   - The New Attribute window is displayed, as shown in Figure 3-1.
3. Enter Name and Display name. Select Attribute type and Character type. Optionally enter Character length.
4. Click Apply and then Cancel.
When creating an attribute, you have to investigate the expected values for that attribute. For example, if you expect the value of an attribute to contain alphanumeric characters, then you assign the attribute a “variable character” attribute type. Furthermore, since this is a “variable character”, you need to decide the maximum and minimum length for attribute value.

There are several types of attributes allowed in Content Manager:

<table>
<thead>
<tr>
<th>Attribute type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>Alphanumeric characters stored at a fixed length</td>
</tr>
<tr>
<td>Variable character</td>
<td>Alphanumeric characters stored at a variable length</td>
</tr>
<tr>
<td>Short integer</td>
<td>Whole numbers between -32767 and 32767</td>
</tr>
<tr>
<td>Long integer</td>
<td>Whole numbers between -2147483647 and 2147483647</td>
</tr>
<tr>
<td>Decimal</td>
<td>Decimal values</td>
</tr>
<tr>
<td>Double</td>
<td>Double precision floating point numbers</td>
</tr>
</tbody>
</table>
Date          dates stored in YYYY-MM-DD format
Time          Times stored in HH:MM:SS format
Time stamp    Timestamps for the application
BLOB          Binary large objects
CLOB          Character large objects

**Note:** By default, Content Manager Library Server only supports up to 320 KB for the CLOB and BLOB attributes. The total amount of character or binary data that can be passed to the Library Server can be increased with some DB2 commands to for example 5 MB. See Technote:

/trs20002.htm

Each character attribute requires 2 additional bytes in the buffer, and the buffer used for binary data also contains control information. In practice, the total amount of application data should be limited to less than 5 MB for each of these attributes. If you need to use large attributes, consider using objects in the Resource Manager.

Attributes can have multiple values and versions. Refer to “Child components” on page 36 for a discussion of multi-valued attributes. See “Versioning” on page 52 for discussion of versions.

The System Administration Client stores these defined attributes and makes them available for selection when you create or modify item types. Refer to “Item type” on page 43 for a discussion on item type.

When creating attributes, try to make them as basic as possible so that they are flexible enough to use throughout your system. Sometimes, some of the same attributes always go together. For these attributes, you can create an attribute group as discussed in the following section.

**Tip:** The *ICMSTATTRDEFS* table contains the definitions for all attributes.

### 3.1.2 Attribute groups

An *attribute group* is a set of attributes that are grouped together for convenience.

When you add an attribute group to an item type (see “Item type” on page 43 for a discussion of item type), all attributes in the attribute group are inserted into the item type at one time.
For example, instead of inserting four attributes for every item type to create an address (street, city, state, and postal code), you can create an attribute group called Address that includes these four attributes, as shown in Table 3-3.

Table 3-3  Sample attribute group, Address, for a student

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Street</td>
</tr>
<tr>
<td>John</td>
<td>100 Almaden Rd.</td>
</tr>
<tr>
<td>Mary</td>
<td>555 5th Ave.</td>
</tr>
<tr>
<td>Richard</td>
<td>239 1st Street</td>
</tr>
</tbody>
</table>

To create an attribute group, use the following procedure:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Attribute Groups from the left-hand tree menu.
2. From the file menu, choose Selected → New. Alternatively, right-click Attribute Groups and select New.
   The New Attribute Group window is displayed, as shown in Figure 3-2.
3. Enter Name and Display name. Select existing attributes from Available Attributes. Click Add to add the selected attributes to the Group Attributes.
4. When finished adding all the attributes, click Apply and then Cancel.
3.1.3 Components

A component is the building block used to form the hierarchical tree of data for each item. There are two types of components, root and child. You can build item types by using one root component and zero or more child components.

**Tip:** The ICMSTATTRGROUP table contains the definitions for all attributes groups.

**Tip:** The ICMSTCOMPDEFS table contains the definitions for both root and child components. In addition, each component has its own table.
**Root components**

A *root component* is the first level of an item type. It consists of both system and user-defined attributes.

For example, a Student item type has a root component that includes Item ID and Component ID as system-defined attributes, as well as Last Name, First Name, and Student Number as user-defined attributes; see Figure 3-3.

![Figure 3-3 Sample root component for student](image)

This root component should contain all attributes (or attribute groups) that apply directly to the item, and for which only one value is expected. If an attribute, or logical collection of attributes is expected to have multiple values, a child component should be created.

**Child components**

A *child component* is the optional second (or lower) level of the hierarchical item type. Each child component is directly associated with the level above it. Use child components for attributes (or sets of attributes) where multiple values may exist.

For example, a student attends one to many classes, and plays several sports. Both classes and sports should be implemented as child components. Figure 3-4 demonstrates the root and the child component definition.
Figure 3-4  Student item with one root and two child components

**Note:** A child component links to its parent component via the Parent ID of the child and the Component ID of the parent. The parent component can be either root or child component. Both Parent ID and Component ID are system defined attributes.

There are no limits to the number of levels in an item hierarchy. If you plan to use the Windows Client or eClient that are provided, be aware that these clients only display first-level child components. If you plan on developing your own client, you can extend your item hierarchy to any number of levels.

For example, in any particular class, a student may take many examinations. Figure 3-5 shows two-level child components for the student item.
You create child components when defining the item type. Refer to “Item type” on page 43 for details.

**Note:** Although there are no limits to the depth of your hierarchy, keep in mind that each child component lies within a separate table. The data retrieval of an item may involve a large number of table joins. In complicated data model hierarchies, the usage of child components may impact performance significantly.

For more information on performance considerations for data modeling, refer to *Performance Tuning for Content Manager*, SG24-6949.

You create child components when defining the item type. Refer to “Item type” on page 43 for details.
3.1.4 Objects

In Content Manager, an object is any data entity that is stored on a Resource Manager in digital form. For example, objects can be JPEG images, MP3 audio, AVI video, and plain text files.

Some file formats are supported natively by Content Manager:

- Microsoft Word
- Lotus WordPro
- TIFF
- JPEG

Objects are managed by items on the Library Server. The items contain the necessary information for describing and locating the objects. Using the items, users can create, retrieve, update, or delete objects.

**MIME type**

MIME type is an Internet standard for identifying the type of object that is being transferred across the Internet. MIME types include many variants of text, audio, image, and video data.

In Content Manager, when you create an object, you specify its MIME type. When an object of that type is retrieved from the Resource Manager, your application reads the MIME type and determines how to handle the object. For example, if the MIME type for an object is GIF, your application may launch a Web browser to view the object.

Adding a MIME type to Content Manager is necessary when documents, images, photos, and other objects added to Content Manager do not have a predefined default handler. This includes identifying the different extensions or suffixes that represent the objects. All files or objects stored in Content Manager need to have their MIME type identified so Content Manager clients and other applications correctly handle the object.

To add a MIME type, use the following procedure:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → MIME Type from the left-hand tree menu.
2. Select from the file menu, select Selected → New. Alternatively, right-click on MIME Type and select New.
   
   The New MIME Type window is displayed, as shown in Figure 3-6.
3. Enter Name, Display name, MIME type and Suffix. Select Valid function. Optionally, enter Application name and Application flags.
4. When finished entering all the information, click **Apply** and then **Cancel**.

![New MIME Type](image)

**Figure 3-6  Create a MIME type**

**Media Object Class**

The **media object class** is used to specify system actions that can be performed on core object types stored on Resource Managers. When defined, the object is associated with a defined attribute group allowing the object class to inherit the system defined attributes for the object type. The Java class is included, along with the specific server side DLL or shared object for handling the object. A media object class needs to be defined before the object type can be associated to an item type. The predefined media object classes in Table 3-4 are sufficient to handle most Content Manager implementations.

**Table 3-4  Predefined media object classes**

<table>
<thead>
<tr>
<th>Media object class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D KlImage ICM</td>
<td>Specific object class handler used for Binary Larger Objects (BLOB) stored on a Resource Manager. This class has been <strong>deprecated</strong> and is provided for compatibility with prior releases.</td>
</tr>
<tr>
<td>Media object class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DKLobICM</td>
<td>Represents an abstraction for a generic large object (LOB) that is stored on a Resource Manager and pointed to by an item on the Library Server. Use DKLobICM to add, retrieve, update, and delete generic Resource Manager objects. To work with more specific types of data, you can use one of the more specific subclasses of DKLobICM: DKStreamICM, DKTextICM, and DKVideoStreamICM.</td>
</tr>
<tr>
<td>DKStreamICM</td>
<td>Represents generic streamable data that is stored on a Resource Manager and pointed to by an item on the Library Server. Use this class to: Add, store, or update large streamable objects from external sources using protocols such as FTP. The adding or storing of objects can be synchronous or asynchronous. Retrieve (synchronously or asynchronously) large streamable objects to external destinations. Specify where to begin and end streaming. Retrieve information about stream duration, rate, format, and group. This class is a subclass of DKLobICM.</td>
</tr>
<tr>
<td>DKTextICM</td>
<td>Represents text data that is stored on a Content Manager Version 8 Resource Manager and pointed to by an item on the Library Server. You can make a DKTextICM object text searchable by indexing the content of the object. This class is a subclass of DKLobICM.</td>
</tr>
<tr>
<td>DKVideoStreamICM</td>
<td>Represents streamable video data that is stored on a streaming server Resource Manager (in this case, IBM Content Manager VideoCharger) and pointed to by an item on the Library Server. Because the content of DKVideoStreamICM objects is often large, you should complete add, update, and retrieve operations through third-party servers using a standard protocol such as FTP. After you retrieve the item from the Library Server, you can use this media object class to initiate a session to stream the content between the video server and player. This class is a subclass of DKLobICM and inherits its methods from the DKStreamICM class.</td>
</tr>
</tbody>
</table>
In addition to the predefined media object classes, you can define your own media object classes as shown in Figure 3-7.

1. Start the System Administration Client. Start the System Administration Client. Expand Library Server (ICMNLSDDB) → Data Modeling → Media Object (XDO) Classes from the left-hand tree menu.

2. Select from the file menu, select Selected → New. Alternatively, right-click on Media Object (XDO) Classes and select New.

   The New Media Object (XDO) Class window is displayed, as shown in Figure 3-7.

3. Enter all the necessary information.

4. When finished entering all the information, click Apply and then Cancel.

![New Media Object (XDO) Class](image)

*Figure 3-7  Create a Media Object (XDO) Class*
3.1.5 Item type

An item type is a template for defining and locating items. It consists of one root component, zero or more child components, and a classification. The template that you use to create specific items is the item type. By using the same template, items of the same type are consistently constructed, which helps you to locate them and quickly define new ones. In Content Manager, you build item types for storing a consistent set of information about related items that you want to catalog.

For example, you have an item type called Student. The Student item type includes a consistent set of characteristics, or attributes, such as Last Name, First Name, Student Number, and Date of Birth. When you create an item of type Student, you enter values for each of these attributes, and the values uniquely define this particular item.

Content Manager comes with a number of predefined item types, as shown in Table 3-5.

Table 3-5  Predefined item types

<table>
<thead>
<tr>
<th>Item type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOINDEX</td>
<td>Default item type used by the Content Manager clients for importing and scanning objects. Item type contains source, user ID, and timestamp attributes. It also contains the following document parts required by the Content Manager clients: ICMANNOTATION, ICMBASE, and ICMNOTELOG.</td>
</tr>
<tr>
<td></td>
<td>Access control is public read.</td>
</tr>
<tr>
<td></td>
<td>Library Server tables: ICMUT01000001 and ICMUT01001001</td>
</tr>
<tr>
<td>ICMSAVEDSEARCH</td>
<td>Default item type used by the Content Manager clients to handle queries.</td>
</tr>
<tr>
<td></td>
<td>Access control is public read.</td>
</tr>
<tr>
<td></td>
<td>Library Server tables: ICMUT01002001 and ICMUT01003001</td>
</tr>
<tr>
<td>ICMFORMS</td>
<td>Default item type for form overlays used by the Content Manager clients. This resource item type has a single variable character extended alphanumeric attribute and uses a Media Object Class of DKLobICM.</td>
</tr>
<tr>
<td></td>
<td>Access control is public read.</td>
</tr>
<tr>
<td></td>
<td>Library Server table: ICMUT01005001</td>
</tr>
</tbody>
</table>
In addition to the predefined item types, you can define your own item type as follows:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Item Types from the left-hand tree menu.
2. Select from the file menu, select Selected → New. Alternatively, right-click on Item Types and select New.
   The New Item Type Definition window is displayed.
3. Enter all the necessary information.
4. When finished entering all the information, click Apply and then Cancel.

<table>
<thead>
<tr>
<th>Item type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMDFOLDERS</td>
<td>Default item type that is typically used by the eClient to route multiple work packages. To use this item type, add the following property to the eClient IDM.properties file: automaticRoutingFoldersEnabled=true. Access control is public read. Library Server tables: ICMUT01005001 and ICMUT01006001</td>
</tr>
</tbody>
</table>

Tip: The ICMSTITEMTYPEDEFS, ICMSTCOMPDEFS, Item Root, and Component system tables are used to build complex objects. When a user defines a new item type, Content Manager inserts a row into the ICMSTITEMTYPEDEFS table. The ItemTypeID is a unique sequential number to be managed by the system. One item root table and either zero or many item child tables will be created.

When item types are created in Content Manager, they need to be classified to assist Content Manager in determining the purpose for the item type. The classification of an Item is a design-time description, which may support specialized functionality, attributes, and behavior.

Tip: The ICMSTITEMTYPEDEFS table contains the values of item type classification. The value in the ItemTypeClass column represents the classification (0 - Item, 1 - Resource Item, 2 - Document, 3 - Document Part).

Content Manager provides four default item type classifications: item, resource item, document, and document part.
Item
The classification of *item* is the default classification. It represents items in the system that do not have associated parts, such as documents, images, photos, or files. The attributes of the item type are sufficient to describe the purpose of the item and to represent the content for the item. In essence, all data is stored in the Library Server database without any associated parts in the Resource Manager.

The benefit of having an item type classification such as item is in the support of integrated solutions. Independent software vendors and solution providers can use this item type classification to define information used in their solution where the data is contained totally within the database tables used by the Library Server. For example, a keyword item can be created to manage a keyword list.

For each item (or folder), the Library Server database has:
- A row in the item type table, and where applicable, items in the child component tables
- A row in the links table for every link, or “contains relationship” it has
- A row in the document routing tables for every process in which it is participating

**Note:** Links and folders are discussed further in 3.1.8, “Relationships (links, auto-linking, references, foreign keys)” on page 59.

Returning to our earlier “Student” example, if we classify a Student item type as an item, we can include its various attributes, but we cannot store any associated objects such as a photo of the student with that item.

Figure 3-8 shows the New Item Type Definition window when using the item classification.
Notice how, in Figure 3-8, the Media Object (XDO) Class field is disabled, and the Document Management tab and the Default Storage tab are also disabled. This is because the item classification does not allow for any associated objects, and therefore no configuration is needed in these areas.

**Note:** This classification is not supported in the Content Manager clients that are provided.

**Resource item**

The classification of *resource items* represents items in the system that *do* have associated parts such as documents, images, photos or files stored on a Resource Manager. The attributes of the item type should describe or clarify the stored resource files.

The benefit of having an item type classification such as this is in the support of integrated solutions. Independent software vendors or solution providers can use the resource item type classification to build compound document management or Internet solutions where an application stores reference data (meta data) in the Library Server linked to the objects stored in the Resource Manager.
For each resource item, the Library Server database has:

- A row in the item type table, and where applicable, items in the child component tables
- A row in the media object class table for that part type
- A row in the document routing tables for every process in which it is participating

Returning to our earlier “Student” example, if we classify a Student item type as a resource item, we can define its various attributes, and optionally can store, for example, a photo of that student with the item.

Figure 3-9 shows the New Item Type Definition window when using the Resource Item classification.

![Figure 3-9   Create an item type (resource item)](image)

Notice how, in Figure 3-9, the Document Management tab is grayed out, but the Media Object Class field and the Default Storage tab are enabled, which allows the administrator to configure the item for the object that this resource item will allow.
In general, if your application utilizes the item and resource item model, it is more efficient. The document model (as described in the following section) requires intermediate tables to hold the links between the document and the parts of which it is comprised. This means that there are additional tables that must be updated when documents are added or changed and these tables must be traversed when retrieving data, as compared to item and resource item. All these additional tasks relate to the costs in processing time on the Library Server and there are consequent performance advantages if they can be avoided.

**Document**

The *document* item type classification allows an item to contain multiple document parts. A document part is essentially a content file but can take other forms as described later, in “Document part” on page 50. The document management section allows multiple parts to be selected and linked to the item when creating an item type. A document item type can be created using the default parts or no associated parts. For example, to represent a folder that can be used with the IBM Content Manager clients, create a document item type with attributes and no associated document management parts. This does the same thing as creating an item type of item; but an item type of item cannot be used with the provided Content Manager clients. If a document item type does have associated parts, they are managed in a parts list, which is a hidden child component of the document item type.

It is important for integrated solutions that share content stored in Content Manager with the Content Manager clients to follow the item type document model. For example, a front end batch capture solution could store the scanned images as ICMBASE parts in an item, based on the document item type. Doing this allows the attributes stored for the item to be used in searches by the Content Manager clients and the images stored as base parts to be displayed in the clients. In cases where the front end batch scan system needs to index information that has been OCR’ed (Optical Character Recognition) from the images, the OCR’ed text files can be stored as an ICMBASETEXT part. The indexed information can then be used to locate the item. Similarly, using the ICMBASESTREAM for storing videos provides a nice way of storing videos so the Content Manager clients can locate the videos and then launch a defined video player to handle the videos.

For each document, the Library Server database has:

- A row in the item type table, and where applicable, items in the child component tables

**Note:** This classification is not supported in the provided Content Manager clients. However, if using the federated eClient, resource items *are* supported.
A row in the parts table for that item type
A row in the media object class table for that part type
A row in the document routing tables for every process in which it is participating

Returning to our earlier “Student” example, if we classify a Student item type as a Document, we can include its various attributes, and can store any number of content files with that item. For example, we can store a photo of the student, an image file containing the student’s academic transcript, and a text file containing the essay that the student wrote when applying to the school.

Figure 3-10 shows the New Item Type Definition window when using the document classification.

![New Item Type Definition](image)

*Figure 3-10  Create an item type (document)*

Note that, in this case, the Media Object (XDO) Class field and the Default Storage tab are not enabled. The Document Management tab is enabled to allow the administrator to add any number of document part types to this item type definition. See Figure 3-11 as an example of this case.
The Document Management tab as shown in Figure 3-11 allows you to add the
document part types, specifying storage and versioning options.

**Note:** Although a document item type is not required to have associated parts,
a document item type must have at least one associated base part, even if it is
empty, to be displayed in the eClient.

**Document part**
Table 3-6 contains a list of predefined document management parts.

<table>
<thead>
<tr>
<th>Document part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMBase</td>
<td>The fundamental part of the document item type used to store documents, images, photos, or any object stored in the system. Library Server table: ICMUT00300001</td>
</tr>
<tr>
<td>ICMBaseText</td>
<td>Similar to the ICMBase part, used for storing textual type documents or files that are intended to be indexed for full text searches. Library Server table: ICMUT00301001</td>
</tr>
<tr>
<td>ICMBaseStream</td>
<td>Similar to the ICMBase part, used for storing videos that can be used with Content Manager VideoCharger. Library Server table: ICMUT00302001</td>
</tr>
<tr>
<td>ICMNoteLog</td>
<td>Used to store the information added to the item notelog in the Content Manager clients. Library Server table: ICMUT00303001</td>
</tr>
</tbody>
</table>
In some implementations, the predefined document management parts are not sufficient. Content Manager provides the capability to define additional document management parts using an item type template.

To create a document part item type, the document part classification is selected and then the desired Media Object (XDO) Class has to be assigned to represent the document part. Once this is done, the document part item can be saved and then be used as an associated part when creating item types based on the document classification.

Figure 3-12 shows the New Item Type Definition window when creating the document part.

<table>
<thead>
<tr>
<th>Document part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMAnnotations</td>
<td>Used to hold the markups (sticky notes, color highlights, stamps, or other graphical highlights) added to objects in the Content Manager when using the Content Manager clients. Library Server table: ICMUT00304001</td>
</tr>
</tbody>
</table>
In this case, most tabs and fields are not enabled. The document part item type needs to be associated with a Media Object (XDO) Class; this is similar to a resource item.

**Note:** If you create a custom defined document part and associate it with your document item type, it is not supported in the provided Content Manager clients.

**Versioning**

In Content Manager, you can keep multiple versions of items and objects. When you create an item type, you can specify the versions for items of that type on the Definition page of the New Item Type Definition window.

You can set different version policies. See Table 3-7.
Table 3-7  Item versioning policies

<table>
<thead>
<tr>
<th>Versioning policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always create</td>
<td>Creates a new version of the item whenever it is updated. Client users are unaware that additional versions are being created until the next time that they retrieve the item.</td>
</tr>
<tr>
<td>Never create</td>
<td>Updates a single stored item every time. Does not keep various versions of the item.</td>
</tr>
<tr>
<td>Prompt to create</td>
<td>Allows client users to decide whether they want to create a new version when they are updating an item.</td>
</tr>
</tbody>
</table>

If you set the version policy to allow multiple versions, you can set a maximum number of versions or allow an unlimited number. If you set a maximum number, and the specified maximum number is reached, Content Manager automatically deletes the oldest saved version when saving the next new version.

The version policy that you set on the Definition page applies to attribute values. For example, if you set a version policy to allow multiple versions of items, and a user changes the value of the Last Name attribute from Doe to Smith, then, Content Manager creates a new, updated version of the item.

If the item type that you are creating is classified as a resource item or document part, the version policy also applies to the object on the Resource Manager. If the item type that you are creating is a document, you can specify supplemental version policy information for the specific document parts.

To specify version policy, use the Define Document Management Relations window, accessible by clicking Add in the Document Management tab. See the sample panel shown in Figure 3-13.
There are three version policies you can specify for the document parts, as shown in Table 3-8.

**Table 3-8 Version policies for document parts**

<table>
<thead>
<tr>
<th>Version policy</th>
<th>Description</th>
</tr>
</thead>
</table>
| No             | Never create.  
                | Do not allow multiple versions of the selected document part. |
| Yes            | Always create.  
                | Create a version of the selected document part whenever that object is edited. |
| User choice    | Prompt user.  
                | Users decide whether to update the version they are editing or store the updates in a new version. |
As mentioned earlier, the version policy for the document part supplements the version policy on the Definition tab. For example, on the Definition tab, you set a maximum of three multiple versions. In the Define Document Management Relations window, you can specify No for the base part, and Yes for the notelog and annotation parts. In this case, one version of the base part and up to three versions each of the notelog and annotation parts can exist at any given time.

**Tip:** The ICMSTITEMTYPEDEFS table contains the version options. The value in the VersionControl column represents the classification (0 - Never create, 1 - Always create, 2 - Prompt). The VersionMax column contains the maximum number of versions.

**Item type subset**

An item type subset is a restricted view of the attributes added to an item type. Users using the Content Manager clients can be restricted by using the subset for access to the attributes in the defined item type. This gives an application the capability to block sensitive data from users that may not have a need to access the information.

For example, you create an item type to use for employee data. All employees have access to an employee’s location and phone number; but only an employee’s manager or upper managers have access to the employee’s salary history. You need to create an item type subset so the regular employees and the other managers can view only the information to which they have access, and see only the portion of it that interests them.

When defining a subset for an item type with root and child levels, at least one attribute from the root level must be assigned to the subset before an attribute from the child level can be assigned. This is the case with each subsequent child level.

Added to the subset support in Content Manager Version 8 is the ability to also filter information based on the values of the attributes the user is allowed to view. This filters the rows of data returned to the user when doing an attribute search. There can be only one filter per component type, and the only supported filter condition is equality. Using the previous “Employee” example, you can set it such that the manager can only view the salary history of the employee whose employee_special = “No”. In this case, if the employee_special = “Yes”, even the employee’s manager cannot view the salary history.

If a component is filtered at one level, levels below that level are filtered as well, but not levels above it. There is a performance impact for using row-based filters, especially when performing complex queries that access several component types that have row filters.
**Item type definition table**

Now that we have seen how to create item types from the System Administration Client, we will take a brief look at what happens in the database regarding the item type creation.

The *item type definition table* (ICMSTITEMTYPEDEFS), *component definition table* (ICMSTCOMPDEFS) *items table* (ICMSTITEMSnnnsss - where nnn is LibraryID and sss is SysSegmentID) and *component tables* (ICMUTnnnnnsss where nnnnn is the ComponentTypeID and sss is the SegmentID) are used to build the complex object of an item type.

When a user defines a new item type, a number of things happen in the Library Server tables:

- A row is inserted into the ICMSTITEMTYPEDEFS table. The row contains an ItemTypeID which is a unique sequential number to be managed by the system. The ItemTypeID is used as a unique identifier for the item type in all associated tables in Figure 3-14.

- A number of rows are inserted into the ICMSTCOMPDEFS table for a root and either zero or many descendant (children) tables.

- A table with a base view is created to store the attribute values for items that the item type stores. The name of the view is maintained in the ComponentViewName column in the ICMSTCOMPVIEWDEFS table. The table is also referred to as the *component table* (ICMUTnnnnnsss, where nnnnn is the ComponentTypeID and sss is the SegmentID). For example, you can see the data of the attributes in this table.

- A number of rows are inserted into the *item type relation table* (ICMSTITEMTYPEREL) for defining relationships between any two item types. It is used to model document to part relationships using references from a child component. For example, there is a row for each document part in the item type.

- The *item auto link table* (ICMSTITEMAUTOLINK) predefines a container of items in one or more item types. For example, if you set 1 on the AutoLinkEnable column in the ICMSTITEMTYPEDEFS table, then when an item of the item type is created, the item will be automatically added to the predefined container or containers.

- A row with the name of the item type is defined in the ICMSTNLSKEYWORDS table for NLS support. You can also see names for attributes, attribute groups, privileges, and ACLs.

Figure 3-14 shows the Library Server tables for the item types and illustrates how they are related through the ItemTypeID.
Figure 3-14  Library Server tables for item type

The Techdoc at the following URL contains two PDF files that display the relationships between the system control tables used by Content Manager V8.3. These PDFs illustrate the Library Server tables and Resource Manager tables:


For more detailed information about the system control tables, see the Content Manager V8.3 Information Center:

http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0

3.1.6 Items

An *item* is a generic term for an instance of any item type, regardless of item type classification. For example, you have item types called Student and Teacher. Each student or teacher that you create is generically referred to as an item.

Depending on the item type classification that you use when you create the item type, the item can be:

- An item, which is self-contained and does not describe or represent an object on the Resource Manager. An item contains information that does not directly equate with an object. For example, if you look up a subject keyword, the resulting item may be a list of items that further narrow the subject, or simply a long textual explanation.
A resource item, which describes and connects to an object on the Resource Manager. If an object is a discrete piece of digital content, an item is a representation of that object. The item is not the object, but it thoroughly identifies the object and “knows” how to find it.

A document or a document part, each of which is an element of the document model. The system recognizes a document as an item and a document part as a resource item.

**Tip:** An item is stored across any number of tables, one for the root component and one for each child component of the item type. These tables have the name *ICMUTnnnnnnnssss*, where nnnnn is a numerical representation of the ComponentTypeID and sss is a numerical representation of the SegmentID.

### 3.1.7 Semantic type

A *semantic type* is a descriptive attribute that assists applications in identifying the behavior (semantics) for specific types of items. For example, a document item type without parts can be used to represent a folder, and a document item type with parts can be used to store documents. Integrated solutions and the Content Manager clients use the semantic types to correctly classify a newly created item. By doing this, the nature of the items can be distinguished in queries performed by the Content Manager clients and integrated solutions.

You specify the semantic type when you create an item, and the semantic type is stored as an attribute value. There are seven predefined semantic types, as shown in Table 3-9.

<table>
<thead>
<tr>
<th>Semantic type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>Refer to the base part for an item.</td>
</tr>
<tr>
<td>Document</td>
<td>Describe a document with base parts (ICMBASE) that may or may not have annotations and notelogs.</td>
</tr>
<tr>
<td>Folder</td>
<td>For use in handling folders containing items or other folders. A folder is a metaphor for a folder in real life, in the sense that it has its own attributes but can be used to contain other items.</td>
</tr>
<tr>
<td>Container</td>
<td>Generic reference used for handling items that contain parts.</td>
</tr>
<tr>
<td>Annotation</td>
<td>Represent the annotation parts in Content Manager.</td>
</tr>
<tr>
<td>History</td>
<td>Provide for migration from earlier Content Manager systems to handle the history log.</td>
</tr>
</tbody>
</table>
In general cases where Content Manager is used as an enterprise document management or imaging system, the predefined semantic types are sufficient and are not directly exposed to users. To define your own semantic types, use the following procedures:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Semantic Types from the left-hand tree menu.

2. Select from the file menu, select **Selected** → **New**. Alternatively, right-click on **Semantic Types** and select **New**.

   The New Semantic Type window is displayed, as shown in Figure 3-15.

3. Enter values for Name and Display name.

4. Click **Apply** and then **Cancel**.

<table>
<thead>
<tr>
<th>Semantic type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>Describe the notelog of information maintained as part of an item using the document model.</td>
</tr>
</tbody>
</table>

**3.1.8 Relationships (links, auto-linking, references, foreign keys)**

Relationships between items in Content Manager can be established in four ways depending on the requirements:

- By linking
- By auto-linking
- By reference
- By foreign key

Table 3-10 outlines the advantages and limitations of each relationship approach.
Table 3-10  Advantages and limitations of links, references, and foreign keys

<table>
<thead>
<tr>
<th>Relationship type</th>
<th>Components related</th>
<th>Related elements can be deleted?</th>
<th>Limited by version?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Root → root</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Auto-linking</td>
<td>Root and/or 1 child → root</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reference</td>
<td>Root or child → root</td>
<td>Specify when creating reference</td>
<td>Specify when creating reference</td>
</tr>
<tr>
<td>Foreign key</td>
<td>Root or child → different item type or external table</td>
<td>Specify when creating the foreign key</td>
<td>Specify when creating the foreign key</td>
</tr>
</tbody>
</table>

**Link**

A link is a bi-directional association between one root component to another root component. The link relationship are between selected common attributes defined at the root level of each item type. Attributes defined at the child levels (components) in the item type cannot be used for link relationships. Using link relationship avoids duplicating the resources in the linked item type when needed. Links are directional. There is a source item and a target item.

**Tip:** The links for an item are stored in a table called ICMSTLINKSnnnsss, where nnn is a numerical representation of the ComponentTypeID and sss is a numerical representation of the SegmentID.

For example, you have students and classes. A class contains zero to many students. A student takes zero to many classes. You need to create a Student item and a Class item, and you want to associate the two. Instead of making Student a child component of Class or vice versa, you associate the two by using a link. In this case, you define a link, and the APIs create an entry in the links table to link the two items, as shown in Figure 3-16.

![Figure 3-16 Link sample between root to root](image)

As illustrated in Figure 3-16, the link is separate from the linked items. It is in a link table that contains information about which linked item is the source, which is the target, and the type of link.
As a default, Content Manager provides two default link type definitions:

- **DKFolder** (Folder contains): Default used for foldering to mimic the connection between a physical folder and document contained in a folder.
- **Contains** (containment relationship): A link where each item’s resource appears to be contained in the linked item. When links are established, the resources for each item type are still managed as independent entities. The link table in the Library Server database maintains the relationship between the linked items.

It is possible to create a symbolic link type that more adequately suits your data model. For example, you may want to create a link type that does not imply containment. To do this, use the following procedures:

1. Start the System Administration Client. Expand Library Server (**ICMNLSDB**) → **Data Modeling** → **Link Types** from the left-hand tree menu.
2. Select from the file menu, select **Selected** → **New**. Alternatively, right-click on **Link Types** and select **New**.

   The New Link Type window is displayed, as shown in Figure 3-17.
3. Enter values for Name and Display name.
4. Click **Apply** and then **Cancel**.

   ![New Link Type window](Figure 3-17)

   **Figure 3-17   Create a link type**

Another important point to consider is that links are more dynamic than references or foreign keys (discussed in detail in the following sections) because links can be added to an item by a user at “run time.”

**Note:** Links have been added for exploitation within integrated solutions built on the Content Manager base. Links are not exposed through the provided Content Manager clients other than for folders. This needs to be considered when creating item types for use with the Content Manager clients.

Due to the open and non-restrictive nature of links, they can contain a cyclic reference back to themselves.
Auto-linking

Content Manager provides auto-linking. Previously, auto-links were established between source and target item types at a root level only and with a single attribute only. In Content Manager V8.3, you can set up multi-level and multi-attribute auto-linking.

With auto-linking, you can set up attribute and attribute group associations across item types so that when data is entered in the attribute or attribute group of one item type, it is also entered into the matching attribute or attribute group of another item type. The data type of the attribute can be character, variable character, integer, and small integer.

As you create item types, you can establish auto-linking to automatically link related item types. You cannot establish auto-linking with an item type that does not exist.

Auto-linking can be at the root or child component level, or both. Any items that are created using the specified item types are automatically linked. If an item of one of the auto-linked types does not exist, it is automatically created. For example, if you create a form that must auto-link with a folder that does not yet exist, the folder item is automatically created.

A root-to-root link is not required when creating a child-to-root link. It is not necessary to select at least one attribute on the root component when defining the auto-linking between root and child components. You can have the link defined with all of the link attributes from the child only.

One or multiple attributes can be linked, and one item type can map to an arbitrary number of item types. When using the “folder contains” link type for auto-linking, add the auto-link rule to the item type that is the content of the folder. Set the Linked to field to the item type of the intended folder.

If we use our students and classes again as an example, then we have an item type for students and one for class. We can create an auto-link from attribute ClassName and ClassYear on Student item type to attribute ClassName and ClassYear on Class item type. When we import an image to the Student item type with attributes StudentName, ClassName, ClassYear, then a folder of item type Class will be created containing the image document. See Figure 3-8.
To enable auto-linking:

1. **Optional**: On the Auto-linking page of the New Item Type Definition notebook, select the check box, *Only show available matching attributes and groups*, to ensure that only attributes and attribute groups at the same level are displayed.

2. Select an item type from the *item type to be linked to* list. A list of attributes and attribute groups for that item type displays.

3. Select attributes or attribute groups from the *current item type* list and *item type to be linked to* list. The following rules apply when set up auto-linking:
   - You cannot link date, timestamp, or time attributes.
   - You can link only required attributes. You specify if an attribute is required on the Attributes page of the New Item Type Definition notebook.
   - You can create links between root and child components of different item types. If you have a link from root to root and child to root, the minimum cardinality must be greater than 0.

4. From the *link type list*, select a link type to associate the attributes or attribute groups. All links between the item types must be the same type.

5. Click **Add** to create a link set and add the attributes to the *associated attributes for link*.

6. From the *item type to be linked to list* under *associated attributes for link*, select an item type. All attributes from this item type that are linked to the current item type display in the *associated attributes for link* list.

7. **Optional**: To delete a link or change the link type, select the linked attribute in the lower table and click **Remove**. You can then recreate the link, as needed.
8. **Optional**: If you have a long list of linked attributes, you can use the **Move up** and **Move down** buttons to order or group the links together while viewing. After item types are linked based on your auto-link definitions, the link remains even if you change the definition.

---

**Figure 3-19**  *Defining auto-linking*

---

**Important**: To enable auto-linking, attributes on both item types involved in the auto-link rule must be set to required. This requirement is to prevent run-time errors if you create a document that automatically creates a folder with no attribute value.
References

A reference is a single direction one-to-one association between the root or child component of an item to the root component of another item as illustrated in Figure 3-20.

In this case, the reference is an attribute that is part of the source item. A reference is actually an attribute group which includes system-defined attributes that define the connection.

To create a reference, use the following procedures:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Reference Attributes from the left-hand tree menu.

2. Select from the file menu, select Selected → New. Alternatively, right-click on Reference Attributes and select New.

   The New Reference Attribute window is displayed, as shown in Figure 3-21.

3. Enter values for Name and Display name.

4. Click Apply and then Cancel.
Once created, the reference can be included in the attribute listing of the source item type, similar to any other attribute. When this is done, the deletion rule of the reference can be selected from one of the following options:

» No action: When the target item is deleted, the referenced item is left alone.
» Restrict: The target cannot be deleted when it is referenced by the source.
» Cascade: When the target is deleted, the source is also deleted.
» Set Null: When the target is completed, the source is set to null.

The source and target that you select help to determine whether the reference is version-independent or version-specific.

**Note:** References have been added for use within integrated solutions built on the Content Manager base. References are not exposed through the Content Manager clients. Keep this in mind when designing your data model.

There is often a requirement to have multiple references of a particular type. For example, an article can have one to many authors. You can implement this relationship using references to reduce data redundancy. In this case, create a child component for the article item type which contains the reference attribute to the author item type. This way, your article can have multiple references to author items.

**Foreign keys**

A *foreign key*, supplied by DB2 Universal Database, is a column or a set of columns in a table that refers to a unique key or the primary key of the same or a different table. A unique key of a database is a column or a set of columns for which no values in a row are duplicated in any other rows. You can define one unique key as the primary key for a table. Each table can have only one primary key.
You use a foreign key to enforce referential integrity among tables. In Content Manager, you can define foreign keys to another item type or to a database table that is not part of the Content Manager system. For example, you have a database table that contains salary information. The database table is not part of the Content Manager system, but you do have an item type in Content Manager for employee data. You can create a connection between the employee data item type and the salary information table with a foreign key, such as employeeID.

Note that when defining a foreign key to another item type, only the attributes of that item type that have been specified as unique appears in the drop-down box provided.

To define a foreign key, use the following procedures:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Item Types from the left-hand tree menu.
2. From the right-hand navigator, select the item type you want to create the foreign key. Right-click on the item type and select Properties. The Item Type Properties window is displayed.
3. Click the Foreign Keys tab.
4. Click Add to open the Define Foreign Key window. See Figure 3-23.
5. Enter Constraint name as a tie between the attributes.
6. Select one of the following options for the Update rule field:
   - Restrict: Target cannot be updated.
   - No action: Target can be updated.
7. Select one of the following options for the Delete Rule field:
   - Restrict: Cannot delete the target when it is referenced by the source.
   - Cascade: When the target is deleted, the source is also deleted.
   - No action: Deleting the target has no affect on the source.
   - Set null: Deleting the target sets the source to null.
8. Select a root or a child component from the Select source component list.
9. In the Select target item type or table field, select one of the following options:
   - Use Content Manager item type: The source and target attributes appear in the Source attributes and Target attributes lists. Select attributes and click Add to pair the source and target attributes.
   - Use external table: The source attributes appear in the Source attributes list. Select an attribute and type and a column name in the Target column field and click Add to pair the source attribute and target column.
10. **Optional**: Select the option **Show target data as dropdown in client** to have the target information displayed in the client.

11. Click **Apply** and then **Cancel**.

**Figure 3-22** Create a foreign key

### 3.1.9 Database indexes

Database indexing is vital to Content Manager performance, particularly as the number of items stored in the system increases. When you define a new item type, the appropriate indexes on the new tables are automatically created.
Indexes on user-defined item attributes are not generated automatically. You must define them manually. For the attributes that are frequently used for regular searches, defining indexes on these attributes improves response time and reduces Library Server resource usage for queries over these attributes. You can choose one or a combination of attributes to create an index and you can specify the order of the index, either ascending or descending. Use combinations of attributes to create an index where users typically use that combination of attribute values to find items in the item type.

To create a database index, use the following procedures:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Data Modeling → Item Types → Item Type Name → Database Indexes from the left-hand tree menu.

2. Select the entry with the same name as the item type. From the file menu, select Selected → New. Alternatively, right-click on the entry with the same name as the item type and select New.

   The New Database Index window is displayed, as shown in Figure 3-23.

3. Enter Name for the database index. Select from the Available attributes, click Add to add the selected attribute to the Assigned attributes. Select the Ascending order or Descending order for DB2 storage/retrieval field for the attribute. Repeat adding all the attributes for the index.

4. When finished adding all the attributes, click Apply and then Cancel.

![Figure 3-23 Creating a database index](image-url)
Content Manager cannot anticipate the way your application chooses to search for the data. Application queries are embedded in the compiled programs as Dynamic SQL, which allows the system to calculate the access path at run time. This means that the database optimizer calculates the best access path to the data for your queries when they are run. The database starts to use your indexes for queries after the indexes are created, assuming that the optimizer chooses to use them. The optimizer may decide that the indexes do not help resolve the queries because they are on the wrong columns or because the optimizer is working without dated statistical information about the table.

It is always a good idea to run `runstats` soon after you create an index so that the optimizer has the best information. If you do run `runstats`, it is also a good idea to `rebind` the database packages as well. The optimizer may be able to use the new index to improve access speed for other SQL and may improve other access paths based on the new statistical information.

Unless there is a major performance problem affecting most users at the time, it is best to create indexes and to run `runstats` and `rebind` at times where user activity is low. These utilities can have a negative impact on server performance while they are running.

For more information on administrative tasks, refer to Chapter 18, “Maintenance” on page 471.

### 3.1.10 Text indexes

Text searching and indexing is another important part of the data modeling process. For more information, see Chapter 5, “Text indexing and searching” on page 117.

### 3.2 The Content Manager meta model

Now that you have an understanding of all the modeling entities that can be used in Content Manager Version 8, it is a good idea to quickly review how they all fit together.

Figure 3-24 shows the Content Manager V8 meta model, outlining the relationship between the data modeling entities.
Chapter 3. Data modeling

Figure 3-24  Content Manager Version 8 meta model

Note that:

- A root component and child component share a common set of “Component” characteristics (inheritance).
- Links only associate root components to root components.
- References can relate any component (root or child) to another root component.
- A child component can have its own child components.
- Document parts and resource items share common characteristics with the root component. (inheritance)
- Resource objects are only associated with resource items or document parts
3.3 Comparison with earlier versions

There have been a significant number of changes in the data model of Content Manager Version 8, both in terms of constructs and nomenclature. Table 3-11 maps the terminology from earlier versions to Version 8.

Table 3-11 Terminology map

<table>
<thead>
<tr>
<th>Content Manager earlier versions</th>
<th>Content Manager 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key field</td>
<td>Attribute</td>
</tr>
<tr>
<td>Index class</td>
<td>Item type</td>
</tr>
<tr>
<td>Part</td>
<td>Document part / resource item</td>
</tr>
</tbody>
</table>

In addition to these changes, many enhancements have been made. These are discussed in the following sections.

3.3.1 Hierarchical item type

In earlier versions of Content Manager, item types, which were called index classes, consisted of a single level. In Content Manager Version 8, item types are composed of a root component and zero or more child components.

You can create a hierarchy of child components, any number of levels deep and with multiple child components at each level. Each child component can in turn own other child components, thus forming a composite aggregate relationship; this is a feature new to Content Manager Version 8. These child components replace multi-valued attributes.

When you remove a root, or other parent component, then the related child components are removed as well.

3.3.2 Items

Version 8 introduces the concept of items. An item is an instance of an item type, which follows the template for the hierarchy. Items can be complete or they can point to an object on a Resource Manager. An item that points to an object on a Resource Manager is a resource item. An object is essentially a large object such as a JPEG image, MP3 audio, AVI video, or a text block that a user can store, retrieve, and manipulate as a single unit.
3.3.3 Versioning

In earlier versions of Content Manager, versioning was available for parts. In Version 8, you can define any item to have multiple versions. After the version limit is exceeded, the oldest version of the item is replaced by the most recent.

Versioning involves the whole item hierarchy, starting from the root component. Child components inherit the version of the root. The version of a child cannot be independently changed. Document parts are still versionable, in addition to the versioning of the item.

3.3.4 Links

Earlier Content Manager versions had a limited concept of a link between a folder and one or more documents. In Version 8, a link is a one-to-many association between items at the root component level.

Such linking is also considered to form an aggregate relationship. Link can represent parent-child associations, similar to the relationship of documents and folders in earlier Content Manager. In addition, in Version 8, the link allows this relationship to be more general. A root component that is linked with other items does not own those items; therefore, deleting the root component that is the parent of the link does not result in deletion of the linked-to child items.

3.3.5 References

A reference is a single direction, one-to-one association between items. You can use references between a root or child component and another root component. A reference is represented as a reference attribute in a component.

A component may have several reference attributes, each of which refer to other root components. In contrast to earlier versions, references are now fully maintained by the system.

3.3.6 Attribute groups

Attributes in Version 8 are the same as key fields in earlier Content Manager versions. However, Version 8 extends the concept of attributes by introducing attribute groups.

You can use attribute groups to group related attributes for convenient use when you are creating item types. Instead of individually locating, selecting, and adding individual attributes, you can select them all by selecting the attribute group. You can continue to maintain the individual attributes without altering the attribute group. Attribute groups cannot be nested. Each member of an attribute group cannot itself be a member of another attribute group.
Workflow

In this chapter, we discuss workflow options, Content Manager's Document Routing, WebSphere Process Server, and the circumstances under which each option can be used. Document Routing is part of the base Content Manager functionality. We describe the various components of Document Routing, explain how to implement them, and finally, show how to customize your Document Routing processes using options such as Library Server exits.
4.1 Workflow options

The following workflow options are available: Document Routing from Content Manager and WebSphere Process Server (WPS).

In this section, we introduce these options. In 4.2, “Document Routing concepts” on page 77, we examine Document Routing in more detail.

4.1.1 Document Routing

Content Manager V8.3 contains a workflow system used to perform document based workflow. Referred to as Document Routing, it is part of the standard functionality provided by Content Manager. It is administered through the System Administration Client and is accessible through the Windows client, the eClient, and the Content Manager APIs.

Document Routing is an ideal option if your workflow requirements resemble these criteria:

- Your solution includes Content Manager for Multiplatforms.
- Your process is primarily limited to Content Manager applications (although the system exits can be used to integrate with external systems).
- You wish to route Content Manager content (documents or folders) within your organization.
- The process flow you wish to implement is sequential, parallel, or ad hoc.
- The process flow has line of business, collection points, or branching on specified user action.
- Your process involves conditional routing based on workflow variable values.
- You need a graphical workflow builder.
- You need document routing support in Windows Client and eClient.

Here are some other important things to consider about Document Routing:

- Process creation is done through the System Administration Client.
- Document Routing has production level performance and scalability, with significant customer usage.
- Document Routing has tight integration with the underlying database (DB2 or Oracle) and enables document routing functions such as:
  - Sequential or parallel routing
  - Integration with Content Manager folders
  - Combined document and workflow search
  - Actions and action lists
  - Document collection points
4.1.2 WebSphere Process Server (WPS)

WebSphere Process Server is the IBM strategic process engine. It can be used to model, deploy, and monitor enterprise processes, including complex and long-running processes.

You may want to use this when the features of Content Manager's Document Routing are not sufficient. For example:

- Your process is not document-centric, but you need access to documents at some point in the process.
- Your process activities are heavily integrated with other applications.
- You require business process integration across departments or enterprises.
- Your process involves applications communications workflow.
- You need advanced modelling and monitoring capabilities and the ability to analyze and simulate business processes.

Integration of the Content Manager and WebSphere Process Server environments is done through the Content Manager/WebSphere Process Server Integration Quick Start, a toolkit that provides code, documentation, and samples for working with Content Manager features from a process running in WebSphere Process Server.

The Quick Start shows common interaction points between the two products, such as retrieving documents, retrieving or updating metadata in Content Manager from WebSphere Process Server, or starting a Content Manager document routing process from WebSphere Process Server. The Content Manager/WebSphere Process Server Integration Quick Start is a free download available from the IBM Content Manager support site at IBM.com.

4.2 Document Routing concepts

In earlier Content Manager versions, Document Routing was known as workflow. An instance of Document Routing is called a work package (see 4.2.1, “Work packages” on page 78).

Document Routing processes can contain a number of different work nodes, such as work baskets, collection points, and business applications. (These work nodes will be explained in later sections.) Processes determine the flow of work to complete. When you create the work nodes for a process, focus on the tasks that users need to accomplish. Privilege sets and access control lists determine who has access and who can complete each task in a process.
Document Routing moves documents or folders from one work node to another. A work node is a step within a process at which items wait for actions to be taken by the end users or the applications, or the items move ahead automatically. Each work node belongs to one or more worklists. A worklist contains a list of work packages based on priority or state (such as suspend or notify).

A work package contains the information that a user needs to complete a task. The user is unaware of a work package because the user works on the item it references, not on the work package itself. A work package contains a set of information such as priority, state, resume time, process, and ItemID being routed. Content Manager supports a complex process, allowing you to create processes that determine what route a work package takes based on variables, the actions of the users, or the applications.

You need to create and manage processes. As part of creating processes, you define work baskets, collection points, business applications, actions, action lists, and worklists. Processes reflect your business process. You may force work to the next step in a process, terminate a process, or suspend a process.

You can set conditions to do these tasks automatically, but sometimes you must update these conditions. For example, instead of suspending a document for ten days, you want to suspend it for seven days. To update this task, you must call an API to suspend a process and pass in the suspend time as an input parameter.

The remainder of this section describes each of these document routing components in more detail and how to implement them using the System Administration Client.

4.2.1 Work packages

A work package is a set of information that includes priority, state, resume time, and the item ID of the item being routed. This item can be a folder or a document. It is used to relate an item to a work node. You do not create work packages. Work packages are created by the system with the information from the user who starts a process. The user logs on to Content Manager and proceeds to start a process. Content Manager prompts the user to specify the process, the item ID that uses this process, and the item priority. Content Manager takes this information and creates a work package that proceeds through the process.
4.2.2 Processes

A *process* is a series of steps (work nodes) through which a work package is routed. A process contains at least one start node, one work node, and one stop node. You can use these one-step processes to create ad hoc processes. Processes can have as many steps as you want.

To define a new process, you must have:

- A name for your process
- A predefined ACL
- Optional: One or more work nodes

To define document routing processes in Content Manager V8.3, you need to use the provided graphical workflow builder. The creation of processes is discussed further in 4.2.3, “Defining a process in the graphical builder” on page 96.

You can create a variety of processes. On the following pages we go through some examples of different workflow types. We use the same symbols used in the graphical workflow builder. You can create:

- A serial workflow (Figure 4-1)
- A parallel workflow (Figure 4-2)
- A branching workflow (Figure 4-3)
- A branching workflow with alternate labels (Figure 4-4)
- A branching workflow with a decision point (Figure 4-5)

A *serial process* takes a work package from start to finish without any deviations. See Figure 4-1.

![Figure 4-1 Simple serial workflow](image-url)
In a parallel process, work packages are routed on multiple routes in the same process simultaneously. See Figure 4-2.

Figure 4-2  Simple parallel workflow

You use a split and a join component in the graphical builder to create a parallel process. See 4.2.3, “Defining a process in the graphical builder” on page 96.

You can also create dynamic processes that allow you to direct work through different routes depending on the actions that you specify.

Content Manager provides one default value for moving a work package from one work node to the next: Continue. You can change this value to any others that are meaningful for continuing a work package and/or you can use the values for branching the work packages.

For example, if you want an insurance claim to go from one node to another, you can select Continue as the path it takes. Then, you create a point where the action of the user dictates where the work package goes next. If the insurance claim is approved, it continues on the Continue path. If it is rejected, you can create a path that branches off the Continue path by using Escalate. Figure 4-3 demonstrates this concept.

Figure 4-3  Simple workflow branching
The only default label available is *Continue*. You can create another label by typing it in the label field provided. Your own label appears as one of the choices in the drop-down menu that is displayed in the clients. For example, if you want to branch on the terms *Approve* and *Reject*, then you can type in these labels for each connector branch. See Figure 4-4.

![Simple workflow branching with alternate labels](image1)

*Figure 4-4  Simple workflow branching with alternate labels*

Another option for branching a workflow is a *decision point*. During a process, users might respond to prompts or change document or folder attributes. You can create a decision point that directs work packages to different work nodes depending on the information that users provided during the process, or on the attribute values or properties of the data flowing through the process.

For example, you can use a value of a variable or a value from an attribute in the work package to branch your workflow. If the value of ClaimAmount is more than $500, then branch to ReviewLargeClaim work node; if the amount is equal to or less than $500, then branch to ReviewSmallClaim. See Figure 4-5.

![Simple workflow branching with decision point](image2)

*Figure 4-5  Simple workflow branching with decision point*
**Work nodes**

A work node is a step within a document routing process at which items wait for actions to be completed by users or applications, or through which items move automatically.

The different work nodes you can create are:

- Work baskets
- Collection points
- Business applications

You can set external functions to activate based on whether a work package is entering or leaving a work node, or when the specified overload limit has been reached at the work node. An administrator can also assign an access control list to each work node for additional security control at the work node level.

From a database perspective, a work node is essentially a list of items awaiting processing. Figure 4-6 shows the Library Server tables for a work node.

![Figure 4-6 Library server tables for a work node](image)

A work node table (ICMUT00202001) consists of a root row that defines the work node’s characteristics; as well as child rows in tables for variable definitions (ICMUT00210001), collection resume lists (ICMUT00203001) and work packages (ICMUT00204001) that define the work packages currently located at this work node.
The work package is modeled as a child component of a work node. This child component has two direct child components: *item resume list table* (ICMUT00205001) to hold the item resume list and *container data table* (ICMUT00208001) to hold the container data information.

**Work baskets**

Each step in a process corresponds to a real-world task, such as verifying a record or rejecting an insurance application. *Work baskets* contain work packages. A *work package* contains the location of a document or folder in a database and its priority. A work basket does not perform any actions on the content; rather, it is an indicator of where a work package is in a process. When you assign an ACL to a work basket, you give access to users who can perform the actions on the work packages contained in that work basket. Refer to Chapter 8, “Security” on page 187 for more information on ACLs.

A work basket is more than just a virtual basket that has a pile of work packages stacked in it. You decide what functions a work basket requires to get a work package to where it needs to go. You can specify, through dynamic link libraries (DLLs), what tasks work packages complete upon entering and leaving a work basket. You can also specify what a work package must do when a work basket cannot contain it by using a DLL on the condition that the work basket is overloaded. Your DLLs must reside on the same machine as the Library Server.

At a work basket, you can prompt users to enter values or display values that were set at previous work baskets or collection points. In addition to displaying or storing variable values, you can use variable values to determine which route to take at a decision point later in the process.

To define a work basket, you need:

- A name for your work basket
- A predefined ACL
- **Optional**: One or more variables
- **Optional**: The full location of any DLLs that you plan to use
To create a work basket, do the following steps:


2. Right-click Work Basket and select New.

3. On the Definition panel (See Figure 4-7):
   a. Enter the name of the workbasket in the name field. Ensure that the name that you use is not also being used as the name of a process. You cannot specify as a subprocess any process that has the same name as a work node.
   b. Optional: Enter a description for the work basket. The description will display in the System Administration Client when you view details.
   c. Optional: In the Long description field, enter an extended description (up to 2048 characters) of the work basket. This description displays only in the Work Basket Properties and Copy Work Basket windows. You might use this field to indicate where you use this work basket or what dependencies it has so that you do not modify or copy it without considering the ramifications.
   d. In the Access control list (ACL) field, select an access control list. Only the ones that you defined previously are available. The Library Server checks this ACL when users want to route work packages from this work basket or if users want to suspend or resume work packages.
   e. Optional: In the Action list field, select an action list. Only the action lists that you defined previously are available. If you do not specify an action list, your eClient and Client for Windows users will see only the named routes (specified as connectors) from this work node.
   f. Optional: The overload limit specifies how many work packages can be put in this work node at any particular time. Although the work package is still added to the work node when this number is exceeded, the overload exit specified on the exit routine page will be executed. The exit routine could for example send a notification to administrators. If you do not specify an exit routine, the overload limit has no effect. Set Overload limit to 0, if you do not want to set a limit.
4. **Optional**: On the Variables panel (See Figure 4-8), specify any variables that the user might need to enter while the work package is at this work basket. You can also specify any variables that you want to display to users at this work basket.

eClient or Client for Windows users see only the variables for which you select Display to users. Client users see the value and properties of the variable. The text that you enter in the Prompt text field is displayed to users as the label for that variable. If you select **Display to users**, but do not enter any prompt text, users see the variable value and properties without a label, which might be confusing.

For the Variable panel:

a. Select variable type (character, integer or timestamp), variable name, variable length (only for character) and default value (optional).

b. Select the option **Display to users** if the variable should be viewable from current work basket.

c. Enter text in the **prompt text** field, if the previous option is selected. This is the text that will be used to prompt for user input.
d. For the User input field, you select one of the following radio buttons:

- **Required**: Users must enter a value.
- **Optional**: Users have an option of entering a value.
- **Not allowed**: Users can only view the variables, but are allowed to change the values — for example, to view a variable that was set in a previous work node (use the case-sensitive variable name from the previous work node).

![New Work Basket]

*Figure 4-8 Define work basket variables*

5. **Optional**: On the **Exits Routines** panel (see Figure 4-9), you specify any exit routines that you want to use when entering or leaving this work basket, or when the work basket is overloaded. For each condition:

a. Under the Link library name, type the path and file name of the DLL (Windows) that you want to use. The link library must reside in the same system as the Library Server. The link library code has to be written in C language. For more information on server exits, refer to “Server exits” on page 112.

Sample input: C:\Exits\WXV2MyUserExit.dll.
b. Type the function name that you want to use as the entry point. You need to define a function for every DLL that you specify, or you get an error. The name of your function must begin with the string WXV2 (Workflow eXit Version 2) to differentiate it from functions that you created prior to Content Manager V8.3. The case-sensitive function name that you enter does not require a path or file extension.

Sample input: WXV2MyUserExit.

**Note:** If you specify an exit routine in the Overload fields, ensure that you specified an overload limit on the Definition panel.

![New Work Basket](image)

*Figure 4-9  Define work basket exit routines*

6. Click **OK** to save your work basket. Click **Apply** to save changes and keep the window open. The work basket is identified by name in the graphical builder.
**Collection points**

A *collection point* is a special work node at which a specified folder waits for the arrival of other specified documents or folders. There is no user interaction required at the collection point to continue a work package to the next work node. Once the collection point’s resume list criteria is met, the work package is sent to the next work node.

**Note:** Documents being routed in a process always flow through a collection point without stopping. Folders do the same if the folder being routed is not one of the specified types defined in the resume list criteria.

At a collection point, you can prompt users to enter values or display values that were set at previous collection points or work baskets. In addition, you can use variable values to determine which route to take at a decision point later in the process.

You can specify, through dynamic link libraries (DLLs) and functions, what tasks work packages complete upon entering and leaving a work basket. You can also specify a DLL and function to execute when the work basket has reached a limit that you specify.

To define a collection point, you need:

- A name for your collection point
- A predefined ACL
- **Optional:** One or more variables
- **Optional:** The full location of any DLLs that you plan to use
- A list of required item types to complete a folder
- A folder item type that will contain the item types

A collection point is strictly used in document routing processes. It has nothing to do with the Resource Manager collections.
To create a collection point, do the following steps:


2. Right-click Collection Point and select New.

3. Select the Definition tab (See Figure 4-10).

   a. Enter the name of the collection point in the name field.  
      **Tip:** Ensure that the name that you use is not also being used as the name of a process. You cannot specify as a subprocess any process that has the same name as a work node.

   b. **Optional:** Enter a description for the collection point. The description will display in the System Administration Client when you view details.

   c. **Optional:** In the Long description field, enter an extended description (up to 2048 characters) of the collection point. This description displays only in the Collection Point Properties and Copy Collection Point windows. You might use this field to indicate where you use this work basket or what dependencies it has so that you do not modify or copy it without considering the ramifications.

   d. In the Access control list (ACL) field, select an ACL. Only those that you defined previously are available. The Library Server checks this ACL when users want to work with variables or change the state or priority of work packages in the collection point.

   e. **Optional:** In the Action list field, select an action list. Only those action lists that you defined previously are available. If you do not specify an action list, your eClient and Client for Windows users will see only the named routes (specified as connectors) from this work node.

   f. **Optional:** The overload limit specifies how many work packages can be put in this work node at any particular time. Although the work package is still added to the work node when this number is exceeded, the overload exit specified on the exit routines panel will execute. For example, the exit routine could send a notification to administrators. If you do not specify an exit routine, the overload limit has no effect. Set Overload limit to 0, if you do not want to set a limit.
4. **Optional**: On the Variables panel (See Figure 4-11), specify any variables that the user might need to enter while the work package is at in the collection point. You can also specify any variables that you want to display to users at this work basket.

   eClient or Client for Windows users see only those variables for which you select Display to users. Client users see the value and properties of the variable. The text that you enter in the Prompt text field is displayed to users as the label for that variable. If you select Display to users, but do not enter any prompt text, users see the variable value and properties without a label, which might be confusing. For the Variable panel:

   a. Select variable type (character, integer or timestamp), variable name, variable length (only for character) and default value (optional).

   b. Select the option **Display to users** if the variable should be viewable from current work basket.

   c. Enter text for the prompt text field.
d. For the User input field, select one of the following radio buttons:

- **required**: Users must enter a value.
- **optional**: Users have an option of entering a value.
- **not allowed**: Users can only view the value, but cannot edit anything.

For example, to view a variable that was set in a previous work node (use the case-sensitive variable name from the previous work node), you can select this option.

![New Collection Point](Image)

**Figure 4-11  Define collection point variables**

5. **Optional**: On the **Exits Routines** panel (see Figure 4-12), you specify any exit routines that you want to use when entering or leaving this collection point, or when the collection point is overloaded. For each condition:

a. Under the **Link Library name**, type the path and file name of the DLL (Windows) or so (UNIX) that you want to use. The link library must reside in the same system as the Library Server. The link library code has to be written in C language. For more information on server exits, refer to “Server exits” on page 112. Sample input: C:\Exits\WXV2MyUserExit.dll.
b. Type the function name that you want to use as the entry point. You need to define a function for every DLL that you specify, or you get an error. The name of your function must begin with the string WXV2 (Workflow eXit Version 2) to differentiate it from functions that you created prior to Content Manager V8.3. The case-sensitive function name that you enter does not require a path or file extension.

Sample input: WXV2MyUserExit.

**Note:** If you specify an exit routine in the Overload fields, ensure that you specified an overload limit on the Definition panel.

![New Collection Point dialog box](image)

**Figure 4-12 Define collection point exit routines**

6. **Optional:** Select the **Resume List** tab. A window appears as shown in Figure 4-13. On the resume list panel, specify the number of items required to arrive in a folder before the work package can continue to the next work node:

a. On the **Folder item type** drop-down list, select a folder item type that should be used to collect items in this collection point.
Note: If a folder at a collection point is not defined in the resume list field *Folder item type*, then it will always pass through the collection point.

b. On the *Required item type* drop-down field, select the item type of documents or folders you want to collect at this collection point.

c. In the *Quantity needed* field, type the number of items of the required item type. For example, an auto insurance claim cannot progress unless it receives two estimates of auto damages.

7. Click **OK** to save your collection point. Click **Apply** to save changes and keep the window open. The collection point is identified by name in the graphical builder.

![Figure 4-13   Define resume list for collection point](image)
A business application is a work node that directs work packages to an external business application that you develop. You can also build your application in such a way that it selects the route for the work package to take after the application completes.

You must develop and store the external business application as a DLL (Windows) before you can define it in a work node. The name of the external business application function must begin with the case-sensitive string WXV2 to differentiate it from the external business applications that you wrote prior to Content Manager V8.3. The reason for this is that the interface to the user function has changed for V8.3; Library Server must be able to differentiate older business applications from the new ones so that it knows which parameters to pass.

Your business application can return character values to the document routing process, for example, a claim amount and an approver's name. You can use the business application data structure to pass data (including any work node variable values that the work package carries or the route that it should take upon return) between the Library Server and your business application. This data structure is described in the section “Routing a document through a process” in the IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide, SC18-9679.

To define a business application work node, you need:

- A name for your business application work node
- A predefined ACL
- An existing business application that is a DLL or shared library
- A function that launches the business application

You must know the name of the DLL or shared library and the function that launches it.

To create a business application, do the following steps:

1. Enter a name for the business application in the Name field.

2. Optional: In the Description field, enter a description (up to 254 characters) of the business application. The description that you type here displays in the System Administration Client when you view details.
3. **Optional:** In the Long description field, enter an extended description (up to 2048 characters) of the business application.

4. In the Access control list (ACL) field, select an ACL for the business application node. Only those that you defined previously are available. The Library Server checks the ACL for this work node when users want to route work packages from this work node forward and when users want to suspend or resume a work package at this work node.

5. In the Link library name field, type the fully qualified file and path name of the external business application.

   Sample input: C:\Exits\WXV2claimapp.dll.

6. In the Function name field, type the name of the function that launches the application. The name of your function must begin with the string WXV2 to differentiate it from functions that you created prior to Content Manager V8.3. The function name that you enter does not require a path or file extension.

   Sample input: WXV2claimapp

7. Click **OK** to save your business application. Click **Apply** to save changes and keep the window open. The collection point is identified by name in the graphical builder.

![New Business Application](image)

*Figure 4-14  Define a business application*
4.2.3 Defining a process in the graphical builder

To define document routing processes in Content Manager V8.3, you need to use the provided graphical workflow builder. You can use the graphical process builder to create a complete process flow with work nodes, connectors, decision points, splits, and joins.

**Note:** You can still run existing processes built using Content Manager Version 8.2. You can also open and modify those processes with the graphical process builder to take advantage of V8.3 functionality. New V8.3 processes can only be run on V8.3 of the Library Server.

Figure 4-15 shows the Library Server tables for a process.

![Diagram](image)

*Figure 4-15  Library server tables for a process.*

A process consists of a root row in the *routing process* table (ICMUT00200001), a number of child rows that define the route in the *routing steps* table, (ICMUT00201001) and the *diagram data* table (ICMUT00209001), where the binary representation of the document routing diagram process built by the workflow builder is stored.
Figure 4-16 shows the symbols for the components used to build a process in the graphical process builder.

![Table of Symbols](image)

<table>
<thead>
<tr>
<th>ICON</th>
<th>Explanations</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Start Node" /></td>
<td>Start Node - Starting point of a work node</td>
</tr>
<tr>
<td><img src="image" alt="Stop Node" /></td>
<td>Stop Node - End point of a work node</td>
</tr>
<tr>
<td><img src="image" alt="Work Node" /></td>
<td>Work Node</td>
</tr>
<tr>
<td><img src="image" alt="Collection Point" /></td>
<td>Collection Point</td>
</tr>
<tr>
<td><img src="image" alt="Business application" /></td>
<td>Business application - Line of business</td>
</tr>
<tr>
<td><img src="image" alt="Decision Point" /></td>
<td>Decision Point</td>
</tr>
<tr>
<td><img src="image" alt="Sub Process" /></td>
<td>Sub Process</td>
</tr>
<tr>
<td><img src="image" alt="Split" /></td>
<td>Split</td>
</tr>
<tr>
<td><img src="image" alt="Join" /></td>
<td>Join</td>
</tr>
<tr>
<td><img src="image" alt="Select" /></td>
<td>Select</td>
</tr>
<tr>
<td><img src="image" alt="Connector" /></td>
<td>Connector</td>
</tr>
</tbody>
</table>

**Figure 4-16  Symbols used in the graphical process builder**

To define a process, you need:

- A name for your process
- A predefined ACL
- A start and a stop node
- At least one work node

When you have created your work baskets, collection points, and business applications, you can join these work nodes together to create a process.

You can define a one-step process, or you can create one process with several steps within it.
To create a process, do the following steps:


2. Right-click Processes and select New - Launch Builder.

The New Process properties window shown in Figure 4-17 appears:

   a. Enter the name of the process in the name field.

   b. Optional: Enter a description for the collection point.

   c. Optional: In the Long description field, enter an extended description (up to 2048 characters) of the process.

   d. In the Access control list (ACL) field, select an ACL. Only those that you defined previously are available. The Library Server checks this ACL when users want to start a document or folder on this process.

![Process Properties]

Figure 4-17  Create new process: Process Properties

3. Click OK to save process properties definitions.
4. The next active window is the graphical process builder. Every process must start with a START node and end with one or more END nodes. So these two nodes are always put in the graphical area as shown in Figure 4-18.

![Figure 4-18 New process in graphical process builder](image)

5. In the top left of the graphical builder process window, there is a toolbar of the icons for all the different process components that are available to build a process. To create a component on the white graphical area, click an icon from the toolbar and then click the white graphical area to place the selected component. You can drag the component to a different position and double-click it to get to the properties dialog box for the component.
To add a specific component, do one of the following steps:

- To add a work node, click either the work basket, collection point, or business application icon and click the white graphical area to place the work node. After the work node is placed, a dialog box appears where you can specify to create either a new work basket, collection point, or business application, or you can select previously created work nodes.

- To add a decision point, click the icon from the toolbar and click the white graphical area to add. There is no logic in the decision point. The logic is implemented in the connectors leaving the decision point.

e. To add a sub process, click the icon from the toolbar and click the white graphical area to add. When the sub process is added, a new dialog appears where you can select previously created processes.

f. To add a split or join, click the icon from the toolbar and click the white graphical area to add. Each split node must have a corresponding join node. The split and join nodes are both virtual nodes in that no activity is performed there. The flow directions are controlled by connectors.

g. There will always be a start node when you create a new process. There can be only one start node. The start node is a virtual node, because no activity is performed there.

h. To add a stop node, click the icon from the toolbar and click the white graphical area to add. Every document routing process diagram contains at least one stop node. The stop node is a virtual node, because no activity is performed there.

i. To add a comment, click the icon from the toolbar and click the white graphical area to add. A comment is any additional explanation that you want to add to the process diagram that does not fit in the names that you give to various nodes and connectors. Comments display only in the process diagram; users do not see these comments.

j. To add a connector, click the connector icon from the toolbar, then click the start component and click the end component for the connector.

If you connect work nodes, there are connectors with labels used for a simple forwarding of the work package, or for workflow branching (see Figure 4-3 on page 80 and Figure 4-4 on page 81).

If you connect from a decision point, then there are more options available. (See Figure 4-5 on page 81.) The decision connector should have a name and a description. You can select if the decision should be based on a work node variable, work package property, or an item type attribute (see Figure 4-19).
6. If you have more than one expression evaluation route for this decision point, you can set the precedence for the routes on the Precedence panel. The Precedence panel displays only when you have defined more than one expression evaluation route for a decision point. At run time, the expressions are evaluated in order of precedence. The route for the first expression that is true is followed. If all of the expressions are false, then the Otherwise route is followed.

7. Before saving the process, verify the process by running the verification process. To verify a process, complete the following steps:
   a. Select File → Verify. The Verify window opens and starts verifying the process.
   b. Review the Verification results list for errors or success.
8. Select **File → Save** to save the process. If the process is not verified successfully, then it can only be saved as draft.

Figure 4-20 shows a complete workflow that is taken from the article “IBM DB2 Content Manager document routing, Part 1: A guided tour to process modeling,” which can be found on the IBM Developer Works Web site:


![Figure 4-20 Simple auto claim process example](image)

These are typical steps for creating a process similar to the one in the example:

1. Create all work nodes that are needed.
2. Create a new process.
3. Place all workflow components on the white area in the graphical builder.
4. Connect workflow components with connectors.
5. Sort workflow components and connectors for a better overview.
6. Verify the process.
7. Save the process.

### 4.2.4 Updating a process

You can update a process at any time, even when a process is in use. Any changes that you make immediately affect the process. If you want to make changes to definitions of a work node used in a process, you should open the work nodes from the System Administration Client: Expand Library Server (icmnlSdb) → **Document Routing** → **Work Nodes** and make the changes. You cannot change the definitions of existing work nodes from the graphical builder.

**Note:** Even though you can update or change a running process, we highly recommend that you complete or remove all work packages before applying any changes.
To change a process name, you need to copy it, rename it, and delete the original process. Otherwise, you cannot change a process name, because the process might be in use.

4.2.5 Deleting a process

If you want to delete a process, you must wait until all work packages on the process are complete. You cannot delete a process when it is in use, nor can you prevent anyone from starting a process that you want to delete. You cannot determine when a process is in use because you cannot view who is using the process. You can attempt to delete the process until the system allows you to delete it or you can use the eClient or Client for Windows to view whether there are active work packages on the process that you want to delete.

To delete a process, do the following steps:

2. Right-click on the process name, and select Delete.

4.2.6 Worklists

A worklist controls the selection and presentation of work to a user. Worklists are used to display work packages that are in one or more work nodes. A worklist spans all work nodes and collection points that it is defined to cover, regardless of the processes that the work packages at these nodes are in. In other words, even though a work node may be used in many processes, if a worklist has been defined to include all work packages for this particular work node, all work packages are displayed, regardless of the process they are in.

You need to assign work nodes and collection points to a worklist and give the worklist an access control list (ACL). The ACL filters out the users that can access the work nodes and collection points. The ACL of the work nodes and collection points further restricts access to the work packages in them. For example, an insurance underwriter and an underwriter assistant can have access to the same worklist; but, based on their privileges and the ACL of the work nodes and collection points, the underwriter has a different list of work packages than the underwriter assistant.
Figure 4-21 shows the Library Server tables for worklists.

![Diagram showing Library server tables for worklists]

**Figure 4-21  Library server tables for worklists**

A **worklist** consists of a root row in the *work list* table (ICMUT00206001) and a child row in the *work node list* table (ICMUT00207001) for each work node in the worklist.

To create a worklist, do the following steps:

1. Start the System Administration Client. Expand Library Server (**icmnlsdb**) → **Document Routing**.
2. Right-click **Worklists**, then select **New**.
   
   The New Worklist window appears as shown in Figure 4-22.
3. Enter a name, description, and ACL for this worklist.

4. You can specify how many work packages to display to your user in the worklist. If you decide not to modify any of the default selections, the worklist returns all work packages to which a user has access, based on priority.

   **Optional**: Change the options specified for Selection order, Quantity to return, and Selection filters fields:

   – Select the order to display the work packages in the worklist:
     - By priority: Sort by priority.
     - By date: Sort in ascending order by work packages’ creation dates.

   – Select the number of work packages that are routing documents or folders to which the user has access; these work packages must originate from work nodes that you include in the worklist (on the Nodes panel) and must match the filter criteria that you select:
     - One: Return one work package at a time.
     - All: Return all work packages that meet criteria.
     - Maximum: Limit the number returned. You must specify the limit here.
– Select one or more methods to filter the worklist:

- Filter on notify state: Enables you to specify whether you want users to see work packages that are in notify state or not in notify state.
- Filter on suspend state: Enables you to specify whether you want users to see work packages that are in suspend state or not in suspend state.
- Filter on owner: Filter the work packages by owner.

**Note:** Owner has nothing to do with security or who started a work package on a process. Owner is merely a label that users can assign to a work package for filtering.

5. Select the **Nodes** tab. Select the nodes that this worklist should include. See Figure 4-23.

![Figure 4-23 Create a worklist - Nodes tab](image)

In this example we have included the ReviewSmallClaim and ReviewLargeClaim node. This means that this worklist includes the work packages that are currently at the ReviewSmallClaim or ReviewLargeClaim step of the document routing process. For this example, we would most likely set up an ACL for approvers and assign it to this worklist, so that only the approvers can use this worklist to view work packages that require approval.
4.2.7 Actions

An action specifies how a user can manipulate the work packages at a work node.

You can create your own actions, or use any of the following system-defined actions:

- **CMclient_Start on Process:**
  Users select this action to start a work package on a document routing process.

- **CMclient_Remove from Process:**
  Users select this action to remove a work package that is currently on a document routing process from that process.

- **CMclient_Change Process:**
  Users select this action to remove a work package from one document routing process and start it on another process.

- **CMclient_View Process info:**
  Users select this action to view information about a selected document routing process.

- **CMclient_Continue:**
  Users select this action to move a work package along in the process, either after they have taken another action or instead of taking another action. This action is separate from, and does not require, a route named Continue.

- **CMclient_Suspend:**
  Users select this action to suspend a work package in the document routing process that it is currently on.

- **CMclient_Resume:**
  Users select this action so that a suspended work package can resume moving through the document routing process that it is currently on.
You can create new actions that users can perform during the steps in the process.

Create an action by completing the following steps:

2. Right-click Actions and then New. The New Action window opens.
3. Type a name for your action in the Name field. The name can be up to 32 alphanumeric characters. You cannot change the name after you create the action.
4. Optional: In the Description field, type a description (up to 254 characters) of the action.
5. Optional: Type an alphanumeric name in the Display name field. This name displays to Client for Windows and eClient users as a menu choice, so you should make the name short and meaningful.
6. Optional: In the Shortcut field, type the keys that give users quick access to the action in a custom client. This shortcut also displays in the custom client menu.
   Restriction: Shortcut settings in this field do not apply to the eClient or Client for Windows, only to custom clients.
7. Optional: Select an icon for your action in the Icon field. If you do not know where the graphic file is located or what it is called, click Choose file. Click Preview to see what the graphic looks like.
   Restriction: You cannot select an icon if your Library Server database is on Oracle.
8. Specify the JavaServer™ Pages™ application, link library, or function for this action based on where it will run:
   - Select the client application types that you want to use this action. Web client, desktop client or both.
   - Depending on the application type you selected, you might need to provide information in one, two, or three of the following fields: Application name, Link library name, or function name.
9. Click **OK** to create your action and close the window. Click **Apply** to save the action and keep the window open to create another action.

### 4.2.8 Action lists

An *action list* is a list of actions that a user can perform on work packages. You can assign an action list to a work node (work basket, collection point, or business application) to specify the actions that the user can take at that step in the process.

Consider what actions you want your users to take on the contents of a work package during the document routing process. For example, a claims adjuster can accept a claims form or reject it as incomplete.

If your company is using the Client for Windows or eClient, the actions that you specify will display in the clients as pop-up menu choices. If you do not assign an action list to a work node, the menu choices are limited to the names of the paths that proceed from that work node in the process.
If you choose to apply an action list, it must be a comprehensive list of all actions performed on a work package or its content.

Build an action list from system-defined actions and actions that you created. If you choose to create an action list, you apply it to one or more work nodes in your process. You can create multiple action lists.

To create an action list, complete the following steps:

1. Click **Document Routing** from the tree view in the System Administration Client window.

2. Right-click **Action lists** and then **New**. The New Action List window opens.

3. Enter a name for your action list in the Name field. The name can be up to 32 alphanumeric characters. You cannot change the name after you create the action list.

4. **Optional**: In the Description field, type a description (up to 254 characters) of the action list. The description that you type here displays in the system administration client when you view details.

5. Populate the list of actions on the right. You can select multiple actions by holding the Ctrl key and clicking each action.
   - Add a selected action from the left list to the right by clicking **Add**.
   - Remove an action from the right list to the left by clicking **Remove**.
   - Use the search fields to search for actions to add or remove from a list.
   - Enter the first couple of letters of what you are looking for and click the search button. The search brings you the first instance of your query. Click search again to find the next instance of your query.

6. **Optional**: You can create additional actions by clicking **Create New Action**.
4.2.9 Customization options

When using Content Manager’s Document Routing services, you have three customizing options:

- **Server exits:**
  - On entering or leaving work nodes
  - On a work node overload

- **Java and C++ APIs:**
  - Using management interfaces - DKDocRoutingServiceMgmtICM
  - Using client interfaces - DKDocRoutingServiceICM

- **Java non-visual beans:**
  - CMBDocRoutingDataManagement
  - CMBDocRoutingQueryService

7. When you finish defining the new action list, click **OK** or **Apply**.

![Figure 4-25 Defining an action list](image)
**Server exits**

As described in “Work nodes” on page 82, server exits can be used to customize your Document Routing processes. A document routing user exit routine is a custom programming application (DLL file) that you can create specifically for a work node. You can set a work node to call a specific function in a specific DLL file in the following situations, when:

- A work package is created and started on a process.
- A work package enters a work node.
- A work package leaves a work node.
- The work node overload condition is reached.

When Content Manager calls a user exit routine, you can retrieve the work package from the work package table using the ComponentID in the myExit API.

User exits for document routing in Content Manager V8.3 have a different structure than earlier versions of Content Manager. To pass work package data (including container data) to and from user exit routines, use ICMUSERSTRUCT, defined in the sample header file IBMCMROOT /samples/server/exit/wxv2tue.h. You can include this header when compiling your own function into a DLL file, for example, WXV2MyExit.dll.

There are a number of input arguments you need for ICMUSERSTRUCT before passing it to the server exit function:

- **lUserEvent**
  
  Indicate in which Work Node Event this user exit was called. Values:
  
  - The work package is entering the work node.
  - The work package is leaving the work node.
  - The work package is at a work node, exceeding the overload limit at this work node.

- **zWPCompID[19]**
  
  The work package Component ID passed to this user exit.

- **szWPItemID[27]**
  
  The work package Item ID passed to this user exit.

- **sWPVersionID**
  
  The work package Version ID passed to this user exit.

- **sNumContainerData**
  
  Number of icmcontainer data structs sent to this exit in pContainerDataStructIn.

- ***pContainerDataStructIn**
  
  Contains container data sent to the user exit.
At the time the exit is called, the work package exists in the work package table, and can be retrieved using the work package ID as the identifier of the row in table ICMUT00204001.

There are two ways to use these exits.

First, if the purpose is to update a line of business database, send a notification, or do something else that does not require updates to Content Manager tables and only requires very basic queries, you can write C code in the exit.

Example 4-1 is a simple Document Routing exit that writes to a file when the exit is triggered.

Example 4-1  Simple Document Routing exit (WXV2myExit.c)

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "wxv2tue.h"

extern long WXV2myExit (ICMUSERSTRUCT *pCMStruct)
{
    ICMCONTAINERDATA_STRUCT * pContainerDataStructIn = pCMStruct->pContainerDataStructIn;
    FILE * pFile;
    switch (pCMStruct->lUserEvent)
    {
    case 1: // Case the work package is entering the work node
        pFile = fopen("C:\exits\myExit\WXV2myExit.txt", "aw");
        if (pFile!=NULL)
        {
            fprintf(pFile, "The user exit was successfully triggered...\n");
            fclose(pFile);
        }
        break;
    case 2: // Case the work package is leaving the work node
    case 3: // Case work package exceed the overload limit at work node
        break;
    default:
    break;
    }
    return 0;
}
```
Example 4-2 is a make file example.

Example 4-2  Make file (WXV2myExit.mak)

ALL : "WXV2myExit.dll"

CLEAN :
   -@erase "WXV2myExit.obj"
   -@erase "WXV2myExit.dll"
   -@erase "WXV2myExit.lib"
   -@erase "WXV2myExit.exp"

C=cl.exe
CPP_SWITCHES=/nologo /MT /W3 /GX /O2 /I "../" /D "WIN32" /D "NDEBUG" /D "WIN32"
   /D "_MBCS" /c
LINK32=link.exe
LINK32_FLAGS=kernel32.lib user32.lib /nologo /dll /incremental:no /machine:I386
   /def:"WXV2myExit.def" /implib:"WXV2myExit.lib" /out:"WXV2myExit.dll"
DEF_FILE= "WXV2myExit.def"
LINK32_OBJS= "WXV2myExit.obj"

Example 4-3 is a definition file example.

Example 4-3  Definition file (WXV2myExit.def)

NAME WXV2myExit
EXPORTS WXV2myExit

Make the WXV2myExit.dll with following command:

nmake /f WXV2myExit.mak

The Exit technique can be used, for example, to write the identification details of
a work package (or the underlying documents in that work package) to a file or
database. Another application can then use the Content Manager APIs, take
these details, and complete a secondary processing. One example where this
can be useful is on the completion of a workflow process. The final work node
can trigger an exit that identifies the document that has just completed workflow
in a file or another database.

Another application, on a scheduled basis, can read this file or database and
move the completed documents to another collection on the Resource Manager,
forcing the document to take on a new migration policy. In this way, a document
can be migrated to another storage system (for example, TSM), upon completion
of its workflow.
Second, if access to the APIs is required, the exit starts a separate process which gets a database connection. The transaction is not committed by Content Manager, therefore, its data is not visible in the application. There are two ways to solve this problem:

- The started process can be run asynchronously, with a slight delay to allow the commit to be performed by the API layer. It may be necessary to retrieve more information from the work package table for use in the application, but the query is simple.

- The exit can include a COMMIT call before starting the process, and the call can then be synchronous. If the work package has entered a collection point, the application should not route the work to another location because the stored procedure is expected to process the collection point rules.

The Content Manager tables are documented in the Information Center so it is possible to write SQLs to get the work package details, which includes both the work node and the process. From that information, it is possible to get the name of each and discover whatever information your custom process requires.

**Java and C++ APIs**
The Java and C++ APIs can be used to fully exploit all Document Routing functionality. These APIs allow you to develop customized applications that take advantage of all the Document Routing functionality, some of which may not be directly exposed through the Content Manager clients. For example, you may want to develop an application which can direct work from one ad-hoc process to another.

For more information on the APIs, see Chapter 6, “Application development overview” on page 131. The chapter also directs you to other publications that assist in learning the APIs. In particular, there are a number of well-documented samples that can be of great assistance to a developer:

- SDocRoutingDefinitionCreationICM
- SDocRoutingDefinitionDeletionICM
- SDocRoutingListingICM
- SDocRoutingProcessingICM

**Non-visual Java Beans**
The non-visual Java Beans are useful in building Document Routing-aware Java and Web-client applications. The support of the bean programming model can be used to expose all of the Document Routing functionality inside your custom applications.
In this chapter, we discuss text indexing and searching. We introduce the concept of text indexing and then demonstrate how to create and maintain these indexes for integration with your Content Manager item types. In addition, we discuss how to use these indexes, and mention any performance considerations you should take into account when creating these indexes.
5.1 Implementation

Content Manager includes an internal text search capability. Text search uses the DB2 Version 8.2 Net Search Extender (NSE) for DB2 database or Oracle Text 9.2.0.6 for Oracle database. Text search works across both attributes and document text content. The new text search capability is tightly integrated with the Library Server database; this enables the automatic synchronization of the Library Server data and the text index data. It also simplifies and controls user access to the text index.

You can make attributes, resource items, and documents text-searchable from the System Administration Client, the New Item Type Definition window. You enable text search on the Definition tab only for the resource item and document item type classes. You enable attributes on the Attribute tab.

In this chapter, we discuss text indexing and searching base on DB2 NSE. For Oracle Text, please read the publication *IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332.

Note: Text search is not yet available for Content Manager for z/OS.

5.1.1 Enabling text search

NSE is a pre-requisite for enabling text search for Content Manager with a DB2 database. We recommend that NSE is installed before installing Content Manager. This makes the installation process easier.

The default text search setting is customized during the NSE installation. To view the default settings, use the following command in a DB2 command window for Windows or a similar command for other platforms:

```sql
db2 connect to <database name>
db2 select * from db2ext.dbdefaults
```

The <database name> is the name of Library Server database, the default is icmnlsdb.

If NSE is installed before installing Content Manager, you can set the Content Manager installation program to automatically enable the database for text search. Otherwise, after installing NSE, you need to issue the following command to enable text search:

```sql
db2text enable database for text connect to <database name>
```
The <database name> is the Library Server database name chosen when Content Manager was installed. Typically, this is icmnlsdb. You must issue this command from a user ID with sysadmin authority for that database instance.

To confirm that enabling text search is successful, do the following steps:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → Library Server Parameters → Configurations from the left-hand tree menu.
2. From the right-hand Content of Configurations window, right-click Library Server Configuration → Properties.
3. Select the Features tab. See Figure 5-1. Make sure the Enable Text Information Extender box is checked.

If enabling text search is not successful, or you installed DB2 NSE after installing Content Manager, perform the following steps to enable text search:

1. In the Library Server configuration screen (see Figure 5-1), check the Text Information Extender box.
2. Enter a valid user ID, such as icmadmin, that has the appropriate privileges.
3. Enter password.
4. Click OK.

![Library Server Configuration](image)

*Figure 5-1  Enabling text-searching for Content Manager*

### 5.1.2 Making documents text-searchable

You can enable text search of the content for an item type of classifications Document, Document part, or resource item.

**Note:** For document parts and resource items, you can only enable text searching when the Media Object Class is DKTextICM. If you think you may want to make your item types text searchable, make sure that you choose the DKTextICM Media Object type when the item type is created.
To make an item type searchable, choose the appropriate classification in the item type classification field and check the **Text searchable** box.

**Attention:** For classification, you can only choose it when you are creating the item type. For an existing item type, the classification cannot be modified.

To get there, do the following steps:

1. Start the System Administration Client. Expand Library Server (ICMNLSDB) → **Data Modeling** → **Item Types** from the left-hand tree menu.

2. Right-click the item type that you want to make text-searchable, and then select **Properties**. See Figure 5-2.

![Figure 5-2 Enabling text search for a Document](image-url)
3. Click the **Text Searchable** radio button to make the item type text searchable. If you just click the check box, the text index will use all default settings. If you want to set options for the text index, click **Options**. See Figure 5-3.

4. Click **OK** to save or update the item type after you finished the setting.

**Attention:** The indexing will not occur immediately. When indexing occurs is depended on the options selected for the index. See 5.2, “Administration” on page 127 for more details.

For a document item type, if you want to store an object in a native text format, do the following steps:

1. Go back to the Item Type Properties window shown in Figure 5-2. Click **Options**.

   The Text Search Options window shown in Figure 5-3 appears.

2. Select **ICMfetchContent** as the User-defined function name. If objects are in popular formats such as Word and Word Pro®, select **ICMfetchFILTER** as the user-defined function name.

**Attention:** If you want to store plain text documents, choose **ICMfetchContent**. In addition, the code page of the text index must match the database code page and the code page of the Library Server.

   If you want to store both plain text and other documents, such as Word or PDF, you should also choose **ICMFetchFilter**.

   For more detailed information, refer to *IBM DB2 Net Search Extender Administration and User's Guide*, SH12-6740.

To enable text search for other object types, you need to define your own functions.
5.1.3 Making attributes text-searchable

You can enable text search of attributes when you add the attributes to an item type in the Attributes tab from the Item Type Properties window. Each time you add an attribute of type Character, Variable Character, BLOB, or CLOB, you can make that attribute text-searchable. To make the contents of the attribute text-searchable, do the following steps:

1. Go to Item Type Properties window.
2. Select the Attributes tab.
3. Select the attribute you want to make text-searchable from the Selected attributes and components area. See Figure 5-4.
4. Check the Text searchable box.
5. Click OK.
You can use the default text search parameters or click **Options** from Figure 5-4 to specify text search parameters in the Text Search Options window.

If the item type contained an attribute that is text-searchable, then the user can perform text search for that particular attribute. For example, an attribute, XYZ_ClaimNumber, is of type Character, and you enable XYZ_ClaimNumber to be text-searchable as shown in Figure 5-4. The user can then query for a claim number in a text search using a client application.

### 5.1.4 Defining text search options

When enabling text search for a document or attribute, you can specify the search options by clicking **Options** as mentioned in the previous section. The Text Search Options window is shown in Figure 5-5.
Figure 5-5  Text search options

This window allows you to specify the parameters listed in Table 5-1.

**Note:** If you do not specify these parameters, default parameters are used.

<table>
<thead>
<tr>
<th>Table 5-1  Text search options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section</strong></td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>Index language</td>
</tr>
<tr>
<td>Language code</td>
</tr>
<tr>
<td>Storage options</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>User defined function</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td><strong>Index update settings</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Model definition</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

These parameters are described in more detail in the *IBM DB2 Net Search Extender Administration and User’s Guide Version 8.1*, SH12-6740.
5.1.5 Making documents text searchable on Unicode databases

**Important:** The content described in this section will only be supported in Content Manager V8.3 Fix Pack 2 and later.

Searching documents in multiple languages was not supported prior to Content Manager Fix Pack 2. With Fix Pack 2, the ability to index and search documents in multiple languages is support for Unicode databases. To get this support, the database must be Unicode and the ICMFetchFilter UDF must be chosen for the text index. In addition, the ICMCCSID Library Server environment variable must be set to 1208. This implies that any plain text documents stored in this index must be encoded in code page 1208 (UTF-8). For more details, see the Content Manager Fix Pack 2 readme document.

5.2 Administration

Once text search is implemented, you should update and reorganize text indexes as needed. The index may be updated manually or automatically based on the index options.

5.2.1 Updating the index

To incorporate a newly created or updated item, you need to update its index. There are several ways that this can be done:

- Content Manager includes a sample program that updates and reorganizes the index for you. There are Java and C++ versions of the program and demo to update an index manually. The method to call this application is in the opening lines of the sample code.

The sample Java code can be found at:

```
<CMB root directory>\samples\java\icm\STextIndexUpdateICM.java
```

The sample C++ code can be found at:

```
<CMB root directory>\samples\cpp\icm\STextIndexUpdateICM.cpp
```
You can manually update and reorganize the index. While you can use the Index update settings to control the frequency that the text index is updated, there are times when items are in a queue waiting to be updated. You can run the following command to immediately update the index:

```
db2text UPDATE INDEX myindex FOR TEXT CONNECT TO icmnlsdb USER icmadmin USING password
```

Where:
- `myindex` is the name of the index. If you are unsure of the index name, you can find out by running the following command:
  
  ```
db2 select indexname from db2ext.textcolumns
```

For more detailed information, refer to Content Manager V8.3 Information Center.
- `icmnlsdb` is the name of the default database. You need to substitute the database name if you do not use the default.
- `icmadmin` and `password` are the user ID and password for the Content Manager administrator. Substitute the values accordingly if you do not use the default.

When you add several items to the system administration database and want to search on them immediately, you can do so through the OO API by using:

```
DKDatastoreDefICM.updateTextIndexes(yourComponentTypeID)
```

This forces NSE to load the data from items into the text indexes.

### 5.2.2 Reorganizing the index

If a text column is often updated, subsequent updates to the index can become inefficient. You need to reorganize the index to improve performance. To do this, run the following command:

```
db2text update index myindex for text reorganize connect to icmnlsdb user icmadmin using password
```

Where:
- `myindex` is the name of the index. If you are unsure of the index name, you can find out by running the following command:
  
  ```
db2 select indexname from db2ext.textcolumns
```
- `icmnlsdb` is the name of the default database. You need to substitute the database name if you do not use the default.
icmadmin and password are the user ID and password for the Content Manager administrator. Substitute the values accordingly if you do not use the default.

Alternatively, you can use the OO API to improve your text search performance:

DKDatastoreDefICM.reorgTextIndexes (yourComponentTypeID)

5.3 Using text search

The main difference between text search on attributes and text search on objects is how the content is stored. When you define an attribute to be text-searchable, NSE creates a text index on the column directly. The text index holds information about the text to be searched. This enables users to search text for that attribute. Text search on objects is different.

5.3.1 Searching for object contents

Searching for the contents of objects works a little differently than searching for the contents of attributes. Instead of indexing a column directly, the system uses a reference to the object's location on a Resource Manager. NSE uses the reference to fetch the contents when it creates a text index. A user performing a search does not notice any difference when searching for objects stored in a Resource Manager. A system administrator, however, has to set up a text resource item type view in order for the search mechanism to locate the contents in the Resource Manager. The text search is performed on the resource item type's attribute TIEREF, which refers to the contents stored on the Resource Manager for text search purposes.

There are some useful examples to demonstrate how to search content, attributes, and documents. You can find these examples from IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide, SC18-9679.

5.3.2 Searching for documents

You can perform text search on the contents of document parts. A virtual component type view ICMPARTS is supported in query as a child of every document in the system. The TIEREF attribute under the ICMPARTS component type view refers to the contents of all the text-searchable parts of that document for text search purposes.
5.3.3 Making user-defined attributes text-searchable

You can make your user-defined attributes text-searchable by using the DKAttrDefICM and DKItemTypeDefICM APIs. Default properties of the created text index can be modified by using the DKTextIndexDefICM class. For more information on using the APIs, refer to Chapter 6, “Application development overview” on page 131.

5.4 Performance considerations

The text search engine resides on the Library Server machine creating additional processing, storage, and network load for this machine.

Text indexing requires the text, from a document or attribute field, to be disassembled into text components and then have them indexed and ordered. The index is built and stored on the NSE server in the Library Server. Similarly, search requests involving traversing these indexes to identify the qualifying items are also done in the Library Server.

Being centrally located has implications for large systems with multiple Resource Managers. During indexing operations, a copy of every document that is to be fully text indexed is transferred from the Resource Manager to the Library Server, converted where necessary to a suitable format, and then indexed by the NSE server. Typically, the number of documents in a Document Management style Content Manager system is relatively few; however, the number and the size must be assessed for network impact and index size calculations.

To enhance performance during indexing, consider the following issues:

- Using a VARCHAR data type to store the text attributes instead of LONG VARCHAR or CLOB.
- Using different hard disks to store the text index and the database files. You can specify the directory in which an index is created when defining the text index as part of the item type definition.
- Ensuring that your system has enough real memory available for all data. NSE can optimize text index searches by retaining as much of the index as possible in memory. If there is insufficient memory, the operating system uses paging space instead. This decreases the search performance.
- Leaving the update commit count parameter as blank during text index definition of the item type definition. This is used during the automatic or manual updating of the index. It slows down the indexing performance during incremental indexing. Setting it to blank disables the updating function.
Application development overview

In this chapter, we discuss application development for Content Manager using the connectors. We explain how to install the connectors and cover the variety of options that you have and the suitability of each option. We also discuss some of the important concepts associated with development for a Content Manager Version 8 server.

Note: It is not our intention in this chapter to describe how to develop a Content Manager application.
6.1 Getting started

Content Manager Version 8 comes with a number of Application Programming Interfaces (APIs) that allow you to develop customized Content Manager solutions. Known collectively as the ICM connector, these APIs are an extension of the framework provided by IBM DB2 Information Integrator for Content. To make use of the APIs, you have to install the ICM connector, which is part of the Information Integrator for Content installation.

Once the ICM connector is installed, you can use the ICM connector APIs to build and deploy custom applications that access Content Manager servers. You can also use the APIs to integrate your existing applications into Content Manager servers.

Before attempting to develop a custom application with the Content Manager APIs, we discuss what you should know or do first.

6.1.1 Where the APIs fit in

The main components of the Content Manager system include a Library Server, one or more Resource Managers, and a set of object-oriented Application Programming Interfaces (APIs). To administer a Content Manager system, you use the provided System Administration Client.

The Library Server provides you with flexible data modeling capabilities, secure access to your system, efficient managing of content, and other features. The Library Server manages the relationships between items in the system and controls access to all of the system information, including the information stored in the Resource Managers configured in the system.

The Resource Manager is the component that stores the actual content of binary objects, such as scanned images, office documents, or videos.

The APIs provide applications with access to the Content Manager system. The APIs are available for both Java and C++. Using the APIs, your applications can take advantage of all of the Content Manager functionality, such as data modeling, integrated parametric and text search, and document routing.
The diagram in Figure 6-1 illustrates how the system components fit together. Keep in mind that this is only one implementation of a Content Manager system. In another system configuration, you may have one or more Resource Managers.

![Diagram](image)

Figure 6-1  How the components fit together

### 6.1.2 Installing connectors

To install the Content Manager connectors, you need to run the Information Integrator for Content installation program. Choose the **Connector** option when the installation program prompts you to select a type in the Setup Type window shown in Figure 6-2.
In the Server Connection Selection screen, select **DB2 Content Manager Version 8**, as shown in Figure 6-3.
These options are the only ones you need for developing Content Manager Version 8 solutions. The others are used when you have an Information Integrator for Content server and need to program for other connectors provided by Information Integrator for Content.

At this stage, you want to install the Information Center, as this provides reference material for API development.

6.1.3 Setting up your environment

When setting up your environment, it is important to note that the Java client-server implementation (communication between the client and server via RMI) is not supported for the Content Manager Version 8 connector. For this reason, when developing directly for a Content Manager Version 8 server, you must always use the server package.

Table 6-1 shows the libraries and packages that you use when developing a Content Manager application.

<table>
<thead>
<tr>
<th>Language</th>
<th>Libraries</th>
<th>Packages to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>cmbcm81.jar</td>
<td>com.ibm.mm.sdk.common</td>
</tr>
<tr>
<td></td>
<td>cmbcm81.jar</td>
<td>com.ibm.mm.sdk.server</td>
</tr>
<tr>
<td></td>
<td>cmbicm81.jar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817.lib</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>cmbcm817.dll</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817.lib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817.dll</td>
<td></td>
</tr>
<tr>
<td>C++ (release)</td>
<td>cmbcm817d.lib</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>cmbcm817d.dll</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817d.lib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817d.dll</td>
<td></td>
</tr>
<tr>
<td>C++ (debug)</td>
<td>cmbcm817d.lib</td>
<td>Not applicable</td>
</tr>
<tr>
<td></td>
<td>cmbcm817d.dll</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817d.lib</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cmbicm817d.dll</td>
<td></td>
</tr>
</tbody>
</table>

Environment variables for Java

There are several important environment variables that need to be changed with the Java development.

Windows

Change the following environment variables:

- PATH — Make sure your PATH contains X:\Progra~1\IBM\db2cmv8\dll
  Where:
  X is the drive on which you installed the Information Integrator for Content.
- **CLASSPATH** — Make sure your CLASSPATH contains the following entries:
  - X:\Progra~1\IBM\db2cmv8\lib\xxx
    Where:
    - X is the drive on which you installed Information Integrator for Content and xxx are the jar files (for example, cmbicm81.jar).
  - X:\Progra~1\IBM\db2cmv8\cmgmt
    Where:
    - X is the drive on which you installed the Information Integrator for Content. Note that this directory should contain a file named cmbcmenv.properties

**AIX**
Change the following environment variables:
- **PATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **LIBPATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **LD_LIBRARY_PATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **CLASSPATH** — Make sure it contains:
  - /opt/IBM/db2cmv8/lib/xxx
    Where:
    - xxx are the JAR files, (for example, cmbicm81.jar).
  - /opt/IBM/db2cmv8/cmgmt/cmbcmenv.properties or the directory where you installed this file.

Use the `-qalign=packed` compiler option so that the objects align properly.

**Solaris and Linux**
Change the following environment variables:
- **PATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **LIBPATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **LD_LIBRARY_PATH** — Make sure it contains /opt/IBM/db2cmv8/lib.
- **CLASSPATH** — Make sure it contains:
  - /opt/IBM/db2cmv8/lib/xxx
    Where:
    - xxx are the JAR files, (for example, cmbicm81.jar).
Use the -qalign=packed compiler option so that the objects align properly.

**Environment variables for C++**

There are several important environment variables that need to be changed with the C++ development.

**Windows**

Change the following environment variables:

- **PATH** — Make sure it contains X:\Progra~1\IBM\db2cmv8\dll.
  
  Where:
  
  x is the drive that Information Integrator for Content is installed on.

- **INCLUDE** — Make sure it contains X:\Progra~1\IBM\db2cmv8\include.
  
  Where:
  
  X is the drive that Information Integrator for Content is installed on.

**AIX**

Change or set the following environment variables:

- **NLS path**
  
  - **NLSPATH** — Make sure it contains /opt/IBM/db2cmv8/msg/En_US/%N.
  
  - **PATH** — Make sure it contains /opt/IBM/db2cmv8/lib.

- **LIBPATH** — Make sure it contains /opt/IBM/db2cmv8/lib.

- **INCLUDE** — Make sure it contains /opt/IBM/db2cmv8/include.

### 6.1.4 Setting up WebSphere Studio Application Developer

To get WebSphere Studio Application Developer set up for Content Manager API development, you need to go through many steps. Notice that the following steps are for WebSphere Studio Application Developer Version 5.1:

1. **Create Classpath Variables:**
   
   a. Select **Window → Preferences**, then **Java → Classpath Variables**.

   b. Click **New**. Enter:

   i. Name: CM_COMMON_DIR
ii. Folder: The path to your common shared Content Manager and Information Integrator for Content directory (for example, c:\Progra~1\IBM\db2cmv8\cmgmt)

c. Click New. Enter:
   i. Name: DB2_DRIVER_PATH
   ii. File: The path to your db2java.zip (for example, e:\IBM\SQLLIB\java\db2java.zip)

2. Create a new Java project:
   a. Select File → New → Project.
   b. Select Java → Java Project, then click Next.
   c. Enter a project name (for example, CMJavaProject), click Next.
   d. Select the Libraries tab.
      i. Click Add Variable. Select CM_COMMON_DIR and click OK.
      ii. Click Add Variable. Select DB2_DRIVER_PATH and click OK.
      iii. Click Add Variable. Select XERCFESJAR and click OK.
      iv. Click Add External JARs. Select cmbcm81.jar, cmbicm81.jar, cmbsdk81.jar, from the CMBROOT install directory. Click Open.
      v. Click Finish.

3. Import the Content Manager Java samples:
   a. Select your new project in the Package Explorer, then select File → Import.
   b. Select File System, then click Next.
   c. In the Directory section, click Browse. Navigate to the directory that contains the ICM samples. (For example, E:\Progra~1\IBM\db2cmv8\samples\java\icm\samples\java\icm.) Click OK.
   d. Click Select All.
   e. In the Folder section, click Browse. Select your new project and click OK.
   f. Click Finish.

4. Run the sample connect program:
   a. In the Package Explorer panel, expand your project, then expand (default package).
   b. Select the SConnectDisconnectICM.java sample.
   c. From the file menu, select Run → Run...
   d. Click New, then select the Arguments tab.
e. In the Program arguments section, enter:
   icmnlsdb icmadmin password

   If you do not use the default icmnlsdb as your Library Server database name, substitute it with your own. Substitute your own user ID and your own password if you do not use icmadmin and password to access your database.

f. Click Run.

After running the sample application, you should get the output shown in Example 6-1.

Example 6-1   Output of running sample code, SConnectDisconnectICM.java

===============================================================
IBM Information Integrator for Content v8.3
Sample Program:  SConnect DisconnectICM
===============================================================
Database: ICMNLSDB
UserName: icmadmin
===============================================================
Connecting to datastore (Database 'ICMNLSDB', UserName 'icmadmin')...
Connected to datastore (Database 'ICMNLSDB', UserName 'icmadmin').
Disconnecting from datastore & destroying reference...
Disconnected from datastore & destroying reference.

6.1.5 Working with sample code

Content Manager provides a comprehensive set of code samples to help you complete key Content Manager tasks. The samples are a great source of API education because they provide reference information, programming guidance, API usage examples, and tools.

You can view the samples from the online API Reference in the Content Manager's Information Center. Additionally, the samples are located in the following directories:

- E:\Progra~1\IBM\db2cmv8\samples\java\icm
- E:\Progra~1\IBM\db2cmv8\samples\cpp\icm

Note: You must have selected the Samples and Tools component during the Information Integrator for Content installation in order to have the samples in these directories.
The readme file covers the basic steps for learning how to build an application using the ICM connector with the ICM APIs. The education modules are found inside the sample files. The samples teach you how to use the APIs and show how to build or integrate Content Management into a custom application.

The Getting Started section in the readme file helps you to quickly learn how to complete the following general tasks:

- Data modeling (for details, see Chapter 3, “Data modeling” on page 29)
- Connecting to a server and handling errors
- Defining attributes and attribute groups
- Working with reference attributes
- Defining your data model
- Working with items
- Working with resource items
- Working with folders
- Working with links
- Defining the Resource Manager
- Defining an SMS collection
- Searching for items (for details, see Chapter 7, “Query language” on page 161)

The Reference Index section helps you to quickly find the sample that contains the concept or topic that you are looking for. Every sample is thoroughly documented and provides in-depth conceptual information and an explanation of each task step. Additional information contained in each sample includes:

- Detailed header information explaining the concepts shown in the sample
- A description of the sample file, including prerequisite information and command line usage
- Fully commented code that you can easily copy, customize, and use in your applications
- A utility function that you can use when developing your applications

It is beyond the scope of this book to cover the details on how to build a Content Manager application; rather, we point you to these samples and the following Information Center and documentations for the detailed “how-to” information:
6.1.6 Application development options

When choosing to develop an application to work with Content Manager, you can choose from a number of options. Some of the elements in Table 6-2 are specific toolkits to be used in a particular context, others are complete APIs that enable you to exploit the full range of Content Manager functionality.

Table 6-2 Programming options

<table>
<thead>
<tr>
<th>Language</th>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Object-oriented interface</td>
<td>Lowest level Java interface allowing access to all of Content Manager functionality (mirror image of C++ API)</td>
</tr>
<tr>
<td></td>
<td>Non-visual Java Beans</td>
<td>Useful in building Java and Web client applications. Supports beans programming model such as default constructors, properties, events, serializable. Usable in Java Bean-aware builders.</td>
</tr>
<tr>
<td></td>
<td>Java Viewer toolkit</td>
<td>Visual classes to build document viewers. Non-visual classes (document services) can be used for document conversion and annotations editing. Used in the eClient viewer applet and middle-tier conversion.</td>
</tr>
<tr>
<td></td>
<td>Servlet &amp; JSP™ Taglib</td>
<td>JSP tag library reduces Java code in JSP servlet with actions. Default actions provided for typical operations. Customizable for add and remove actions.</td>
</tr>
</tbody>
</table>
With so many options, the decision should be made early regarding the applications development option(s) that should be used. The following list highlights some of the factors that you may wish to consider:

- **Developer skill level:** Although the low-level APIs in both Java and C++ provide the most powerful functionality, they are also more appropriate for the advanced programmer. The other Java options such as the visual and non-visual Java Beans may be easier to use for some developers, but may not provide the granularity of functionality or flexibility that is required for a particular application.

- **Current investment in resources:** What skills already exist in your organization is a factor to consider. If you currently have many highly skilled C++ developers, then obviously you will most likely want to develop your application using the C++ APIs.

- **Applications required:** If you develop your own customized “thick” client, you can choose either C++, Java, or even the automation objects. If you desire to deliver an application to users through a browser, then the Java options should be your choice.

- **Customization of existing clients:** If you customize the existing Windows client by implementing user exits, you must use C++. Similarly, if you customize the eClient, then you must use the Java API.
For more information, refer to the Information Center and Table 6-3.

**Table 6-3  Further information on programming options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Where to look</th>
</tr>
</thead>
</table>
| Object-oriented interface (Java or C++) | *Application Programming Guide*<sup>a</sup>  
Java code samples  
(E:\Progra~1\IBM\db2cmv8\samples\java\icm)  
C++ code samples  
(E:\Progra~1\IBM\db2cmv8\samples\cpp\icm) |
| Non-visual Java Beans          | *Application Programming Guide*  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\java\beans) |
| Visual Beans                   | *Application Programming Guide*  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\javaim\beans\gui) |
| Java Viewer toolkit            | *Application Programming Guide*  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\java\viewer) |
| Servlet & JSP Taglib           | *Application Programming Guide*  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\jsp) |
| User-exits (C++)               | *Client for Windows Programming Reference*<sup>b</sup>  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\server\exits) |
| Automation (OLE)               | *Client for Windows Programming Reference*  
Code samples  
(E:\Progra~1\IBM\db2cmv8\samples\activex\fed) |

<sup>a</sup> The complete documentation name is *IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide*, SC18-9679.

<sup>b</sup> The complete documentation name is *IBM DB2 Content Manager Enterprise Edition / IBM DB2 Content Manager for z/OS Client V8.3: Client for Windows Programming Reference*, SC27-1337.

### 6.1.7 Understanding the differences

These are the basic differences between the Java and C++ API sets:

- The operators defined in the C++ APIs are not defined in the Java APIs. They are supported as Java functions.
- The Java class object (java.lang.Object) is used in place of the C++ class DKAny to represent a generic object.
Common and global constants are defined in the interface DKConstant in the Java APIs; in C++, they are in DKConstant.h.

The Java APIs use Java’s garbage collector.

The Java functions DKDDO.toXML() and DKDDO.fromXML() are not available in C++.

6.2 Application development concepts

The APIs that implement Content Manager Version 8.3 functionality are grouped into the ICM connector. The ICM connector APIs have an ICM suffix, as in the example DKDatastoreICM.

Both the Information Center and the IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide, SC18-9679, go into great detail about creating a content manager application. Please see these publications for more details and examples demonstrating application development for Content Manager.

In this section, we provide a summary of some of the important concepts to remember.

6.2.1 Understanding components

For conceptual purposes, you can categorize the Object Oriented APIs into the following groups of services:

- Data and document modeling
- Search and retrieve
- Data import and delivery
- System management
- Document routing

The data and document modeling module contains the APIs that enable you to map your business data model to the underlying Content Manager hierarchical data model. For more information on data modeling, see Chapter 3, “Data modeling” on page 29. Figure 6-4 shows the Data Modeling API hierarchy.
The search and retrieve module processes requests for the managed items such as documents and folders. The search module APIs enable you to perform combined text and parametric searches for items contained in the Content Manager system. The search results are returned to the application in the form of search result sets. For more information on searching a Content Manager repository, see Chapter 7, “Query language” on page 161. Figure 6-5 shows the Query API hierarchy.
The data import and delivery module provides the APIs that enable you to import data into your system and deliver that data through various media, such as a network or the Web.

The system management module provides you with the interfaces to configure and maintain an efficient and a secure Content Manager system. For example, you can incorporate the system management APIs into your application to allow you to adjust the system control settings, manage users, assign users privileges, and allow access to the system. Figure 6-6 shows the some of the System Management API hierarchy.
The document routing module APIs help you to route business objects, such as documents, through a process, as defined by your business requirement. See Chapter 4, “Workflow” on page 75 for more information on document routing. Figure 6-7 shows the Document Routing API hierarchy.
6.2.2 Representing items using Dynamic Data Objects (DDO)

A DDO is essentially a container of attributes. An attribute has a name, value, and several properties. One of the most important properties of attributes is the attribute type. A DDO has a persistent identifier (PID) to indicate the location of the object in persistent storage. A DDO has some methods to populate itself, and corresponding methods to retrieve an item’s information. The DDO methods include add, retrieve, update, and delete. You use these methods to move an item’s data in and out of Content Manager.

In memory, Content Manager items are represented as DDOs. Item attributes are represented as DDO attributes with a name, type, and a value. Links and references are represented as special types of attributes. The difference between a link attribute and a reference attribute, however, is that a reference attribute refers to another (single) DDO or Extended Data Object (XDO), and a link attribute refers to a collection (multiple) of DDOs or XDOs. XDOs are used to represent large objects (LOBs).

A reference to an item, either to an XDO or another DDO, has a name with the type property set to object reference, and value set to refer to the instance of the referenced object. Child components and links are also represented as DDO attributes with the type property set to a collection of data objects, and value set
to a collection of DDOs. In the case of a child component, the attribute name is the name of the child component. The value is the collection of child components belonging to the root component. If the root item is deleted, all of the child components of the root item are also deleted.

6.2.3 Working with Resource Manager

A Content Manager Resource Manager controls a collection of managed resources (objects). It also manages the necessary storage and Hierarchical Storage Management (HSM) infrastructure; but you must first configure the Resource Manager to support HSM. Resource Managers have facilities to support type-specific services for more than one type of object, such as streaming, zipping, unzipping, encrypting, encoding, transcoding, searching, or text mining.

A single Resource Manager is used exclusively by the Library Server. Each Resource Manager delivered by the Content Manager system provides a common subset of native data access APIs through which it is accessible by the controlling Library Server, by other Content Manager components, and by the applications, either locally (on the same network node) or remotely.

Other data access APIs allow remote access to a Resource Manager using the Resource Manager's own client support or a standard network access protocol such as CIFS, NFS, or FTP. For remote access, use a client-server connection. Clients communicate with the Resource Managers using HTTP through the use of a standard Web server. Data delivery is based on HTTP, FTP, and FS data transfer protocols.

Using HTTP, any application or a Content Manager component that needs to access the Content Manager-managed content can dynamically form a triangle with a Library Server and Resource Manager. This triangle forms a direct data access path between the application and each Resource Manager, and a control path between the Library Server and the Resource Manager. You can map this conceptual triangle to any network configuration, ranging from a single-node configuration to a geographically distributed one.

The architecture also accommodates Resource Managers that an application is not able to access directly, such as a host-based subsystem, a single-user system that does not handle access control, or a system containing highly sensitive information where direct access by an application is not allowed by business policy. In this case, access to such Resource Managers is indirect. Both the pull and the push paradigms of data transfer are accommodated by the Content Manager system as well as synchronous and asynchronous calls.
Working with Resource Manager objects
Within Content Manager, every managed entity is called an item. Items come in two types, the type that represents pure logical entities such as documents or folders, and the type that represents physical data objects such as the text data of a word processing document, the scanned image of a claim, or the video clip of an automobile accident. Objects have a special state and behavior needed to handle the physical data associated to a logical document.

Resource objects also represent entities such as files in a file system, video clips in a video server, and BLOBs. At run time, resource objects are used to access the physical data they point to. For that reason, resource objects in Content Manager have a type. They have a specific state and behavior. The Library Server and the Resource Manager share a schema to store the state of an object. The base object types provided by Content Manager are: generic BLOBs or CLOBs, Text, Image, and Video content objects. You can also create sub-classes of the pre-defined types. A resource object can also have user-defined attributes, which are used for search and retrieval.

From the Content Manager system perspective, each object is represented by a unique logical identifier, the Uniform Resource Identifier (URI). The Library Server manages the URI name space. On request, the Library Server maps URIs onto Uniform Resource Locators (URL). URLs are used to gain access to the physical data. URLs do not point directly to a storage area managed by the Resource Manager. Instead, the Resource Manager uses a local name space to convert logical object names to physical file names. Object URIs are created by the specific Resource Manager. The Library Server or the end-user can suggest an object URI (its name), but the decision is made by the Resource Manager.

You can access an object using the Content Manager Resource Manager APIs (store, retrieve, update, and delete). In some cases, you can use APIs that are native to the object (stream, multicast, and stage) or the file system.

6.2.4 Working with transactions
Transactions allow Content Manager to maintain consistency between the Library Server and any adjoining Resource Manager. A transaction is a user-determined, recoverable, unit of work, that consists of one or more consecutive API calls made through a single connection to the Library Server. The sequence of consecutive DKDatastoreICM method calls are made either directly or indirectly, through the DDOs and XDOs.

The scope of a transaction and the amount of work within that transaction is by default the work performed by a single API method (implicit transaction). This type of transaction is recommended and is the best performing scope of a transaction. You can, however, change the scope of a unit of work, making it
larger to include multiple method calls (explicit transaction); but using this type of
transaction can introduce performance overhead.

When a transaction ends, the entire transaction is either committed or rolled
back. If it is committed, all of the Content Manager server changes made by API
calls within the transaction are permanent. If a transaction is rolled back or fails,
all the changes made within the transaction are reversed during rollback
processing.

The commit and rollback of a transaction are done automatically in the case of
implicit transactions. In the case where explicit transactions are in use, the
transaction commit is controlled by the application, whereas a transaction
rollback can be initiated by an application or automatically by the Content
Manager system. The Content Manager system initiates a rollback when a
severe error occurs or when it is necessary to resolve a deadlock between the
Library Server and the database.

Within a transaction, uncommitted Resource Manager changes are not visible to
the application that made the changes until the transaction is committed. For
example, you make changes to a Resource Manager item and you store it. If you
retrieve that item before the transaction is committed, the item does not reflect
the changes that you just made. You do not see the updated item until the
transaction is committed.

Concurrent or overlapping transactions through a single Library Server
connection is not supported. To maintain concurrent transactions, you must
make multiple connections between the Library Server and the database.
Applications such as IBM WebSphere Application Server handle processes,
connections, and sessions.

The execute() and executeWithCallback() methods in DKDatastoreICM
automatically create an additional connection to the database when invoked. The
new database connection is then used to execute the query. Since queries use a
separate database connection, they also have a separate transaction scope from
the other content server operations. The connection to the database is closed (or
returned to the pool, if pooling is enabled) when the DKResultSetCursor is
closed.

**Things to consider when designing transactions**

If a client node or the Library Server fails before the transaction is committed, the
database recovery function rolls back the transaction on the Library Server
immediately. The Resource Manager changes made during the failure are
undone immediately if the client node and Resource Manager are both active. If
the client node itself failed, you should put the Resource Manager through a
cycle of the Asynchronous Recovery Utility in order to restore consistency
between the Resource Manager and the Library Server. Before the utility runs, the servers still have data integrity. What is affected are the operations on the in-progress items that had the failure, which will be rejected until the Resource Manager is recovered. Failure during in-progress update of an object prevents another update of that same object, until the first failure is reconciled.

If the Resource Manager fails, you should get a system administrator to run the asynchronous recovery utility to remove inconsistencies. On z/OS, the Resource Manager has native transaction capabilities, such as Object Access Method (OAM), which are used to recover more expediently.

**Caution when using explicit transactions**

For explicit transactions, where you control the transaction scope using `DKDatastoreICM.startTransaction()` and `DKDatastoreICM.commit()`, use caution when developing an application where you work with Content Manager Documents with parts and when performing `DKLobICM create`, retrieve, update, and delete (CRUD) operations. When performing these operations, you should perform CRUD operations as closely as possible to the end of the transaction. You should also keep a transaction as short as possible, since a long transaction increases the potential for database locking problems.

Locking problems are most apparent when updating an item, and the application chooses to not commit the transaction immediately. As long as the transaction is not committed, the item that is being updated, is still visible to other applications. When another user attempts to access or view the item, that user is locked out until the updated transaction is committed. The same problem (database locking) occurs when creating new items in a folder. If the folder is visible to another user, and that user attempts to retrieve the new item, the user is locked out until the transaction is committed. The amount of time prior to the transaction commit is the amount of time the user is locked out.

The best approach to avoid database locking is to commit transactions often and to avoid long running transactions. If you must perform CRUD operations within a transaction, it is recommended that you perform these operations when it is understood that no one else accesses the items being updated.

**Using check-in and check-out in transactions**

Content Manager supports check-out and check-in operations on items. The check-out operation is called to acquire a persistent write lock for items. When an item is checked out by a user, other users can not update it although they can still retrieve and view it. You need to call the check-out operation prior to updating or re-indexing an item, regardless of the transaction mode (implicit or explicit) that you use. When you are done with the item, call the check-in operation to release the persistent lock and make the item available for other users to update.
After you create an item, you have the option to keep it in the checked-out state to prevent other users from changing it until you are completely done with the work. If you check-out (or check-in) an item using an explicit transaction, the checkout is undone if the transaction is rolled back. If you check-out an item using an implicit transaction, the checkout is committed. It is the application's responsibility to check the item back in, using check-in options or methods.

**Processing transactions**

The transaction scope can be controlled by a client API call; but it must be designed carefully. To group a set of API calls into a transaction, you must build it explicitly by completing the following steps:

1. Call the `startTransaction()` method of the `DKDatastoreICM` class. You work with the `DKDatastoreICM` methods to complete all the transaction steps.
2. Call all of the APIs that you want to include in the transaction in the order that you want them called.
3. Call the commit or rollback methods to end the transaction.

All of the API calls made between the `startTransaction()` and either `commit()` or `rollback()`, are treated as one transaction.

All APIs can be included in transactions, unless specifically noted. See the online API Reference for details. Some administrative APIs cannot be included in the explicit transactions. For example, the method to define or update item types cannot be included in the explicit transactions.

Below is the list of class methods involved in Content Manager transactions in relation to item creation and update:

- `DKDatastoreICM.startTransaction()` — Starts an explicit transaction.
- `DKDatastoreICM.commit()` — Commits transaction changes to the database.
- ` DKDatastoreICM.rollback()` — Rolls back or removes transaction changes from the database.
- `DKDatastoreICM.checkOut()` — Acquires a persistent write lock on an item.
- `DKDatastoreICM.checkIn()` — Releases a previously acquired persistent write lock.
- `DKDatastoreICM.add()` — Creates a new item in the database.
- `DKDatastoreICM.updateObject()` — Updates an item. The item must be checked out prior to calling this method.
- `DKDatastoreICM.retrieveObject()` — Retrieves an item from the database.
- `DKDatastoreICM.deleteObject()` — Deletes an item from the database.
DKDatastoreICM.moveObject() — Re-index an item. Moves an item from one item type to another item type. The item must be checked out prior to calling this method.

Another great source for information about transactions is the SItemUpdateICM sample.

6.2.5 Using logging and tracing

This section covers logging and tracing.

Library Server logging and tracing

If you write your own application, it maybe helpful to turn on the performance tracing.

**Important:** If you turn on any tracing, it may affect the performance of your system.

Library Server error information is logged in the ICMSERVER.LOG file. You can modify the default settings for ICMSERVER.LOG through the Content Manager System Administration Client:

1. Start the System Administration Client. Expand Library Server (icmnlsdb) → Library Server Parameters → Configurations from the left-hand tree menu.
2. From the Contents of Configuration panel, double-click Library Server Configuration.
   The Library Server Configuration window opens.
3. Click the Log and Trace tab. See Figure 6-8.
4. Modify the default directory of the ICMSERVER.LOG file if needed.
5. Select the check boxes for the level of trace information you want to log into the log file.
6. Click OK.
You can select four levels of trace information to log into the Library Server log file, ICMSERVER.LOG:

- **Basic**: Entry and exit information for the Content Manager stored procedures and lower-level Library Server functions (for example, list NLS keywords).
- **Detailed**: Basic trace information, plus information on the lower-level controls through the Library Server programming logic. This trace level provides information on how the programming logic run.
- **Data**: Information on what input parameters were passed into the Content Manager stored procedures, and the intermediate data as the stored procedures are running.
- **Performance**: Information on how fast the Content Manager stored procedures run. The trace shows one line for each stored procedure and the elapsed time, in milliseconds, that the stored procedure took to run.

The ICMSERVER.LOG file provides information for problem diagnosis and corrective action by your IBM service representative.
If tracing is requested by a client application, the trace level set by the administrator is the maximum that is allowed. If the administrator sets the trace level to 0, no tracing information is available regardless of application requests. Significant errors are still logged even if tracing is not required.

There is an additional tool in the System Administration Client that enables you to configure additional logging features for debugging and troubleshooting purposes. For example, with Content Manager V8.3, you can now trace a single user’s activities. For more information on logs and traces for Library Server, refer to Chapter 21, “Troubleshooting” on page 559.

**Resource Manager logging**

In Content Manager V8.3, the logging for each Content Manager component can be controlled from within the System Administration Client using the following steps:

1. Start the System Administration Client. Select **Tools → Log Configuration**.
2. Click **Resource Manager** on the left panel (see Figure 6-9).

*Figure 6-9  Log configuration utility - Resource Manager*
From the log configuration utility, you can log the following Resource Manager components:

- Asynchronous recovery utility
- Migration sub-process
- Purge sub-process
- Replicator sub-process
- Resource Manager
- Stage sub-process
- Validation utility

For each component, you can specify the logging level as follows:

- Error
- Warning
- Informational
- Trace (entry and exit)
- Trace (full)
- Performance

For more information on logs and traces, refer to Chapter 21, “Troubleshooting” on page 559.

**API logging**

This section explains how connector logging is activated. The Information Integrator for Content connector logging utilities log all exceptions, including those exceptions that are not errors. Occasionally, error messages may appear in the log file that are not propagated to the end-user. In some cases, the API or the user-application is able to recover or continue in the case of warnings.

**Attention:** When reading the log files, keep in mind the context within which the exceptions and messages are logged.

**Java**

Java has two log managers: default and LOG4J. You can configure and use only one of the log managers at a time. The same configuration file, cmblogconfig.properties, is used to control the type of log manager used and the configuration specific to each type of log manager. For more information about the log manager that you want to use, see the section in the configuration file, cmblogconfig.properties, that pertains to the log manager you want to use.

When the connector logging utility is first instantiated, it searches the CLASSPATH of the Java Virtual Machine instance to find the logging configuration file: cmblogconfig.properties. If this configuration file is not located, the default logging setting is used.
In addition, you can control the logging from within the System Administration Client. From its main menu, select Tools → Log Configuration, and click Java API from the left panel. You can specify the logging level. In addition, you can specify the log file path, log file name, maximum log file size in MB, and maximum number of files. Similarly, you can configure logging for the Java Beans by clicking Beans from the left panel.

Refer to Chapter 21, “Troubleshooting” on page 559 for more logs and traces information.

C++

C++ has one log manager: default. C++ references the same log configuration file as Java does; but the Information Integrator for Content C++ connectors consult only the default log manager logging settings.

- **Windows:** When the C++ connector logging utility is first instantiated, it reads the configuration file cmblogconfig.properties from the directory to which %CMCOMMON% is pointing. By default, %CMCOMMON% points to C:\ProgramData\IBM\db2cmv8\cmgmt. If the configuration file is not located, the default logging settings are used.

- **AIX:** When the C++ connector logging utility is first instantiated, it reads the configuration file cmblogconfig.properties from the directory to which /opt/IBM/cmb/cmgt is pointing. If the configuration file is not found, the default logging settings are used.

In addition, you can control the logging from within the System Administration Client. From its main menu, select Tools → Log Configuration, and click C++ API from the left panel. You can specify the logging level. In addition, you can specify the log file path, log file name, maximum log file size in MB, and maximum number of files.

Refer to Chapter 21, “Troubleshooting” on page 559 for more logs and traces information.

**Working with the logging configuration file**

This section explains how to work with the settings in the logging configuration file of the Content Manager connector, cmblogconfig.properties.

**Default settings**

The cmblogconfig.properties file contains the following default settings. It is important not to change these default settings in case the configuration file cannot be found, or if other errors occur with user-defined settings:

- It uses the default log manager.
- The default log file name is dklog.log.
dklog.log is placed in the current working directory where an Information Integrator for Content-enabled application is run.

- The logging priority is set to Error.
- Maximum number of exceptions of the same error message ID to allow is 5.

**Modifying cmblogconfig.properties**

To update the settings in the cmblogconfig.properties file, follow these steps:

1. If you are using the default installation directories, change the directory to c:\Progra~1\IBM\db2cmv8\cmgmt\connectors or /home/ibm/cmadm/cmgmt/connectors. If you have changed the default installation directories, see the IBMCMMROOT environment variable for the current location of this file.

2. Open cmblogconfig.properties in a text editor.

3. Change the settings to be used by the default log manager:
   - **Section 0 - Global Settings:** Determines maximum exception count.
   - **Section 1 - Log Manager Factory Setting:** Determines whether you use the default log manager or LOG4J.
   - **Section 2 - Default Log Manager Setup:** Section 2 has three subsections:
     - **Section 2.1:** Specify Log Priority. Eight priority settings are available. The default priority setting is Error.
     - **Section 2.2:** Log Output Destination Setting. Three settings are available: Log to a file, Log to Standard Error. The default setting is Log to a file.
     - **Section 2.3 - Log File Name Setting:** Use only when the option in section 2.2 is set to Log to a file. The default log file name is dklog.log. The log file path must include double back slash ( \ \ ) or single slash ( / ) characters in place of single back slashes if any are used. If you do not specify a path, the log always is located in the current working directory where a problem is executed.

4. Save the file.

You can modify the following priority levels in the configuration file, cmblogconfig.properties.

- **DISABLE** — Disables logging.
- **FATAL** — Provides information that the program encountered unrecoverable errors and must cease operating immediately. (Stopping the program is done separately, not from the logging facility.)
- **ERROR** — Provides information whether the program encountered recoverable or unrecoverable errors. The system is still able to continue operating.
- **PERF** — Used to collect output information for measuring performance.
- **INFO** — Provides significant event messages, such as successful logon.
- **TRACE_NATIVE_API** — Used for logging before and after a native call; provides parameters and return data information.
- **TRACE_ENTRY_EXIT** — Used for signaling entries and exits of program modules (or code blocks).
- **TRACE** — Used to output additional diagnostic information, such as program state changes.
- **DEBUG** — Used to output information for debugging errors.

**Attention:** The log manager continues to append log outputs into the existing log file. We recommend that you periodically delete unwanted log output from the log file to prevent the file from becoming too large.

### 6.3 Additional resources

As mentioned in earlier sections, there are many very convenient additional resources that you should refer to when starting on a Content Manager application development project. They are:

- **Information Center for IBM DB2 Content Manager V8.3**: The information center covers all aspects of API development and includes online API reference and samples for all API options. It can be found from the following URL:
  
  [http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0](http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0)

- **Sample Code**: The sample code is an excellent source for learning how to develop Content Manager applications. The readme file lists the examples demonstrating nearly any concept that you need to know.

- **IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide,** SC18-9679: This guide covers, in depth, all aspects of application development using all the Information Integrator for Content connectors.

- **IBM DB2 Content Manager Enterprise Edition / IBM DB2 Content Manager for z/OS Client V8.3: Client for Windows Programming Reference,** SC27-1337.
Query language

In this chapter, we discuss the new query language that provides simple and efficient access to data stored in Content Manager's datastore. We show how the query language supports the full Content Manager data model and demonstrate with examples, how to use the query language to do both parametric and full-text searching.
7.1 Query language overview

Content Manager Version 8 introduces a new XML-based query language that provides simple and efficient access to Content Manager data.

The query language provides the following benefits:

- Supports the full data model.
- Provides support for versions, such as searching for a specific version and for the latest version.
- Enables searches within component type view hierarchies, across linked items and references.
- Combines parametric and text search.
- Provides SORTBY capabilities.
- Enforces Content Manager access control.
- Conforms to XQuery Path Expressions (XQPE), a subset of W3C XML Query working draft, unlike proprietary languages.
- Executes high performance searches.

The query language searches hierarchical item types and locates items quickly and easily. Before you begin to write queries, you must understand the query language concepts, syntax, and grammar.

All queries are done on item type views. The names that you use for root components or child components in your query strings can be either the names used in an item type definition (both the root and child components) or the names of item type subsets that you have created.

When you submit a query for a document, folder, or object, your request is directed to the Content Manager query engine, which processes the query and translates it to the appropriate SQL. The Library Server then checks the security against ACLs and executes the query. Figure 7-1 shows the process flow when executing a Content Manager query.
Content Manager query language incorporates two kinds of searches:

- Parametric search
- Text search

### 7.1.1 Parametric search

Items are often retrieved by initiating a search on selected attributes. A single query can examine both system-defined and user-defined attributes of the items in the content server. Simple search conditions consist of an attribute name, an operator, and a value that are combined into a clause. Content Manager provides comparison operators to complete parametric searches. The operators include:

- `=`
- `<=`
- `>=`
- `LIKE`
- `NOT LIKE`
- `BETWEEN`
You can specify complex search conditions by combining simple search conditions into a clause using the Boolean operators AND, OR, and NOT. Refer to the query examples for more details.

7.1.2 Text search

Using the DB2 Universal Database Net Search Extender (NSE), Content Manager provides two types of text search:

- Text search on attributes that contain text in components
- Text search on objects.

The main difference between the two types of text search is how the content is stored. When you define an attribute to be text searchable, you are indicating that one can search text contained in the column of that attribute. To make an attribute text searchable, NSE creates a text index. The text index holds information about the text to be searched. This information is used to perform text search efficiently.

In the discussion of text search using ICM query language, NSE can be used interchangeably. From the perspective of ICM query language, there is no difference in text search syntax or functionality between NSE. For more information on text indexing and text searching, refer to Chapter 5, “Text indexing and searching” on page 117.

7.2 Understanding query language

There are three important building blocks in query language: symbols, grammar, and escape sequences. To build a search string, you need to understand these building blocks and how to use them.

In this section, we cover these building blocks. In 7.3, “Query strings” on page 172, we provide concrete examples to help reinforce the understanding of these building blocks.

7.2.1 Symbols

Table 7-1 shows the symbols that can be used within a search string, and explains what they are used for.
7.2.2 Grammar

The grammar for query language is listed in great detail in the IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide, SC18-9679. It uses a standardized and widely used notation, called the ISO EBNF notation, to demonstrate the query language grammar for Content Manager Version 8. Refer to the documentation for details.

7.2.3 Escape sequences

To support advanced features of the query language, such as the wildcards “%” or “_” inside of text strings, escape sequences are used to differentiate between the cases when wildcards are treated as regular characters versus the cases when the wildcards are given the special meaning of wildcard characters. For a user, it is important to know which characters are used as wildcards because wildcard characters, when intended to be treated as regular characters, must be preceded by an escape character. Escape sequences are also used to handle single and double quotes.

You need to add escape sequences when the strings used in queries contain either special characters (double-quote, apostrophe) or wildcard characters (percent sign, underscore, star, question mark) or a default escape character (a backslash). This handling is simple for strings used in comparison conditions and it becomes more involved for the LIKE operator and text search functions. Proper

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Indicates a root item or a direct child of a component.</td>
</tr>
<tr>
<td>//</td>
<td>Indicates any descendent of an item such as child and grandchild.</td>
</tr>
<tr>
<td>.</td>
<td>Represents the current component in the hierarchy.</td>
</tr>
<tr>
<td>..</td>
<td>Represents the parent of the current component.</td>
</tr>
<tr>
<td>@</td>
<td>Denotes an attribute. Should be followed by an attribute name.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Denotes a conditional statement or a list.</td>
</tr>
<tr>
<td>,</td>
<td>Separates elements in a list of values.</td>
</tr>
<tr>
<td>=&gt;</td>
<td>De-references and represents linking or referencing action.</td>
</tr>
<tr>
<td>*</td>
<td>Denotes all (wildcard).</td>
</tr>
<tr>
<td>%</td>
<td>Used with LIKE operator. Denotes any sequence of characters (wildcard).</td>
</tr>
<tr>
<td>_</td>
<td>Used with LIKE operator. Denotes any single character (wildcard).</td>
</tr>
</tbody>
</table>
handling of special characters ensures successful execution of queries and correctness of query results.

**Important**: Use wildcard characters in your query sparingly because using them may increase the size of your result list significantly, which can impact performance and return unexpected search results.

Notice that, in this section, we demonstrate the various escape character requirements with a number of examples. Pay attention as how the escape characters are used. We examine the rest of the example content in more detail in 7.3, “Query strings” on page 172.

Table 7-2 summarizes the escape sequence rules presented following the table.

**Table 7-2  Escape sequence summary**

<table>
<thead>
<tr>
<th>Operators</th>
<th>Special characters</th>
<th>Escape sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparison operators</td>
<td>Double quotation mark &quot;</td>
<td>Precede the double quotation mark with another double quotation mark</td>
</tr>
<tr>
<td></td>
<td>Single quotation mark '</td>
<td>Do nothing</td>
</tr>
<tr>
<td>LIKE operator</td>
<td>Double quotation mark &quot;</td>
<td>Precede the double quotation mark with another double quotation mark</td>
</tr>
<tr>
<td></td>
<td>Single quotation mark '</td>
<td>Do nothing</td>
</tr>
<tr>
<td>Wildcards as regular characters % and _</td>
<td>Precede the wildcard character with an escape character, and add an ESCAPE clause with the escape character after the LIKE phrase.</td>
<td></td>
</tr>
<tr>
<td>Wildcards as wildcards</td>
<td>Do nothing</td>
<td></td>
</tr>
<tr>
<td>Escape characters as regular characters</td>
<td>Precede the escape character by itself.</td>
<td></td>
</tr>
<tr>
<td>Basic text search</td>
<td>Double quotation mark &quot;</td>
<td>Precede double quotation mark with another double quotation mark</td>
</tr>
</tbody>
</table>
### Using escape sequences with comparison operators

When using comparison operators such as `=`, `!=`, `>`, `<“, “BETWEEN” and others, you need to escape the following characters:

- **Double quotation mark:**
  
  Precede your double quotation mark with another double quotation mark:

  ```
  //JOURNAL_ARTICLE[@Title = "Analysis of "The CM Implementation and Migration Cookbook""]
  ```

  Since the book title contains the name of the book in double quotes, "The CM Implementation and Migration Cookbook", these internal double quotes need to be escaped.

- **Single quotation mark:**
  
  You do not need to escape in this case.

  ```
  /REVIEW_ATICLE[@Title != "John Smith’ Redbook Review"]
  ```
Using escape sequences with LIKE operator

When using the LIKE operator, you need to escape the following characters:

- **Double quotation mark:**
  
  Precede your double quotation mark with another double quotation mark:
  
  ```sql
  //Journal_Article[@Title LIKE "Analysis of ""The CM Implementation and Migration Cookbook"" %"]
  ```
  
  Since the article’s title contains the name of the book in double quotes, “The CM Implementation and Migration Cookbook”, these internal double quotation marks need to be escaped.

- **Single quotation mark:**
  
  You do not need to escape in this case:
  
  ```sql
  /REVIEW_ATICLE[@Title LIKE “John Smith’ Redbook Review”]
  ```

- **Wildcards (“%”, “_”):**
  
  The percent sign “%” is a wildcard character used to represent any number of arbitrary characters in a string used with the LIKE operator. The underscore “_” is a wildcard character used to represent a single arbitrary character. If you want these wildcard characters to be treated as regular characters, you need to do the following steps:

  a. Precede the wildcard character with an escape character.

  b. Add an ESCAPE clause with the escape character after the LIKE phrase.

  The following example shows how wildcards “%” and “_” are used to find a book whose title is uncertain:

  ```sql
  /Book[@Title LIKE "Plato%s%S_mposium"]
  ```

  The following example contains an underscore as a regular character (not a wildcard), you can escape the underscore with an exclamation point character “!”. Any single character can be used as an escape character to make a wildcard a regular character:

  ```sql
  //Journal_Article[@Title LIKE "Usage of underscore !_ in query" ESCAPE "!"]
  ```

  In the following query, the wildcard characters are used for both regular characters escaped by “\” and as wildcards to catch both upper case and lower case versions of the word “Usage”, as well as “%” to catch multiple endings of the string:

  ```sql
  //Journal_Article[@Title LIKE "_sage of underscore \_ in%" ESCAPE "\"]
  ```
You can also use an escape character as a regular character. To do so, precede the escape character with itself, as in the following example to search for Yahoo!

```xml
//Journal_Article[@Title LIKE "Usage of underscore !_ on Yahoo!!" ESCAPE "!"]
```

**Using escape sequences with basic text search**

When using basic text search, you need to escape the following characters in the contains-text-basic and score-basic functions:

- **Double quotation mark:**
  
  Precede your double quote with another double quote. For example:

  ```xml
  //Journal_Article[contains-text-basic (@Title, "Analysis of ""The CM Implementation and Migration Cookbook"" ")=1]
  ```

  Since the article’s title contains the name of the book in double quotes, "The CM Implementation and Migration Cookbook", these internal double quotes need to be escaped. The book title is enclosed in apostrophes to keep it as a phrase.

- **Single quotation mark:**
  
  Precede the apostrophe with another apostrophe. Basic text search syntax allows terms enclosed within single quotes, so that a term can contain a space. The doubling of the apostrophe is therefore necessary to differentiate the case of an apostrophe occurring within a term from the case of an apostrophe starting a new term.

  ```xml
  //Book[contains-text-basic (@Title, "John Smith'' Redbook Review")=1] SORTBY (score-basic (@Title, "John Smith'' Redbook Review"))
  ```

  Note that, after John Smith, there are two apostrophes.

  In the following example, Plato’’s has two apostrophes and ‘Plato’s Symposium’ is enclosed in single quotes since it is a phrase.

  ```xml
  //Book[contains-text-basic (@Title, " +Greek +'Plato’’s Symposium' -Socrates ")=1] SORTBY (score-basic (@Title, " +Greek +'Plato’’s Symposium' -Socrates "))
  ```

- **Wildcards (‘*’, ‘?’ and ‘\’):**
  
  Precede ‘*’, ‘?’ and ‘\’ characters with a backslash ‘\’ if these characters are not to be treated as wildcards, but as regular characters. Star ‘*’ is a wildcard character used to represent any number of arbitrary characters in a basic text search for the contains-text-basic and score-basic functions. The question mark ‘?’ is a wildcard character used to represent a single arbitrary character.
The following example shows how to use basic text search when the spelling of a term is not certain. The “*” and “?” characters are meant to be wildcards in this case, so they are not escaped:

```
/Book[contains-text-basic (@Title, " +Greek +'Plato*s*S?mposium' -Socrates ")=1] SORTBY (score-basic (@Title, " +Greek +'Plato*s*S?mposium' -Socrates "))
```

In the following example, the title contains the question mark “?” as a regular character, so this character is escaped with a backslash:

```
/Book[contains-text-basic (@Title, "Why forgive\?")=1] SORTBY (score-basic (@Title, "Why forgive\?"))
```

In the following example, each backslash “\” that naturally occurs in the search term "C:\OurWork\IsNeverDone" must be escaped with another backslash.

```
//Journal_Section[contains-text-basic (@Title, "C:\OurWork\IsNeverDone")=1] SORTBY (score-basic (@Title, "C:\OurWork\IsNeverDone"))
```

**Using escape sequences with advanced text search**

When using the advanced text search, you need to escape the following characters in the contains and score functions:

- **Double quotation mark:**

  Precede your double quote with another double quote.

  In the following example, the article’s title contains the name of the book in double quotes, ”The CM Implementation and Migration Cookbook”, these internal double quotes need to be escaped.

  ```
  //Journal_Article[contains-text (@Title, " 'Analysis of "The CM Implementation and Migration Cookbook" %' ")=1]
  ```

- **Single quotation mark:**

  Precede the apostrophe with another apostrophe. A single apostrophe is not allowed in advanced text search because a set of apostrophes is used to enclose a term or a phrase. If an apostrophe appears inside a term, then the apostrophe needs to be escaped to differentiate it from the apostrophe that ends the term or the phrase.

  In the following example, there are two apostrophes after John Smith:

  ```
  /Book[contains-text (@Title, " 'John Smith'" Redbook Review" ")=1] SORTBY (score (@Title, " 'John Smith'" Redbook Review" "))
  ```

  In another example, there are two apostrophes after Plato:

  ```
  /Book[contains-text (@Title, " ('Greek' & 'Plato''s Symposium') & NOT 'Socrates' ")=1] SORTBY (score (@Title, " ('Greek' & 'Plato''s Symposium') & NOT 'Socrates' "))
  ```
Wildcards ("%", "_"):

Just as the LIKE operator, advanced syntax uses "%" and "_" as wildcards. The percent sign "%" is a wildcard character used to represent any number of arbitrary characters. The underscore "_" is a wildcard character used to represent a single arbitrary character. If you want a wildcard character to be treated as a regular character, you need to do the following steps:

a. Precede the wildcard character with an escape character.

b. Add an ESCAPE clause after EACH term where you use the escape character.

In the following example, an exclamation mark "!" is used as an escape character before the underscore:

```
/Book[contains-text (@Title, " 'Usage of underscore !_ in query' ESCAPE '!' ")=1] SORTBY (score (@Title, " 'Usage of underscore !_ in query' ESCAPE '!' "))
```

Note that an ESCAPE clause must be added after every term in your text search string where you escape wildcards, even if the escape character is the same in all the terms.

```
/Book[contains-text (@Title, " 'Usage of underscore !_ in query' ESCAPE '!' | 'Yahoo! For Dummies' | 'Usage of underscore !_ on Yahoo!!' ESCAPE '!' | 'War and Peace' ")=1]
```

**Using escape sequences in Java and C++**

Precede special characters (for example, double quotes and backslash) with a backslash.

Here is a sample query:

```
/Book[contains-text-basic (@Title, "Why forgive\?")=1]
```

With Java, the query becomes:

```java
String query = "/Book[contains-text-basic (@Title, "Why forgive\?")=1]";
```

With C++, the query becomes:

```cpp
DKString query ("/Book[contains-text-basic (@Title, "Why forgive\?")=1]");
```

Notice how the internal double quotes and the backslash before the question mark are preceded by a backslash. This handling is inherent to Java and C++ programming languages. For more information, refer to the specifications for these languages.
7.3 Query strings

The Content Manager query language exploits various features of the data model. To help you better understand the query language and to get you started with writing queries, this section provides you with a sample data model and a number of sample queries.

Figure 7-2 shows the sample data model which is used for all the example query strings shown in this section.

![Sample data model for query examples](image-url)
7.3.1 Basics

When creating a query string, remember that the entire Content Manager datastore is the starting point for any query, and that you narrow down this query by specifying the relevant criteria. At the next level are the items and item types. To specify items of a particular item type, begin the query string with a “/” to indicate a direct descendent of the datastore. To further narrow down the query, specify attribute criteria using a conditional statement (Example: [@ <Attr_Name>=<Value>]).

There are several important points to remember:

- Query strings use the item type and attribute names, not their descriptions.
- Use escape characters when specifying string values.
- Values may contain arithmetic operations to be computed by the query engine before processing.
- Functions such as latestVersion() or DB2 functions may be used as values.
- If versioning is enabled, all versions are returned unless latest version is specified.

**Tip:** It is often useful to read a query string backwards to understand what it is trying to retrieve. Since it is the last component in the path that is returned as the result of the query, this method often simplifies interpreting a query string.

Here are some basic query examples:

- This example finds all journals. The query gets all instances of the Journal item type. The “/” starts at the implicit root of the datastore. Each item type is an element under this root:
  
  /Journal

- This example finds all journals with exactly 50 pages. The predicate @NumPages = 50 evaluates to true for all journals that have the Content Manager attribute NumPages set to 50:
  
  /Journal[@NumPages=50]

- This example finds all journals with the number of pages between 45 and 200. Note, you can perform arithmetic operations to calculate the resulting values to be used with the BETWEEN operator:
  
  /Journal[@NumPages BETWEEN 49-4 AND 2*100]

- This example finds all root components that have a title. To eliminate the restriction that only root components should be returned, the query can be rewritten to start with a double-slash:
  
  /*[@Title]
7.3.2 Multiple item types

Using the OR operator “|”, queries can be extended to search for multiple item types, as shown in the following example:

```
(Book | Journal) [(@/Journal_Author/@LastName = "Smith" OR
   ./Book_Author/@LastName = "Smith") AND (@/Book_Section/@Title LIKE "CM%"
   OR ./Journal_Section/@Title LIKE "CM%")]
```

Alternatively, we can use:

```
(Book[.//Book_Author/@LastName = "Smith" AND .//Book_Section/@Title LIKE
   "CM%"] | (Journal[.//Journal_Author/@LastName = "Smith" AND
   ./Journal_Section/@Title LIKE "CM%"]))
```

The above two queries produce the same result. “./Journal_Author” means that a component Journal_Author should be found either directly under the current component in the path (which in the first case is either a Book or a Journal) or somewhere deeper in the hierarchy. Note that the LIKE operator is used in conjunction with a wildcard character, in this case “%”.

7.3.3 Text search

In Content Manager Version 8, text search queries are incorporated into the common query language. The following sections show examples of both the basic and the advanced text searching in Content Manager queries.

**Basic text search**

Since the majority of text searches are done by simply listing a few words one after the other, basic (simplified) text search syntax was designed specifically to make this most common case easy for users. The syntax also allows for use of “+” and “-”, as well as for use of quoted phrases. Simplified text search is done by using `contains-text-basic` and `score-basic` functions. The `contains-text-basic` function is used to search within attributes or within content of resources or documents. The `score-basic` function uses the same syntax and is used for sorting results based on the rank of the text search results. To check if it is true, equate the `contains-text-basic` function to 1; to check if it is false, equate the `contains-text-basic` function to 0.

Here is some additional information about basic text search syntax:

- Can perform case-insensitive text search (just as in the default case of the advanced syntax). See the NSE documentation for case-sensitive search options.
- Term within a pair of single quotes is assumed to be a phrase.
Uses “+” (plus) and “-” (minus):

- “+” (plus) implies that the document must include this word.
- “-” (minus) implies that the document must not include this word.
- When a “+” or “-” is not specified, the query engine uses an algorithm to match the words to the text.

Boolean operators (AND, OR, NOT) are not valid and are ignored.

Parentheses in the basic syntax are not supported.

Valid wildcards include:

- “?” (question mark) represents a single character.
- “*” (asterisk) represents any number of arbitrary characters.

The following example shows a basic text search using contains-text-basic and score-basic functions:

```
//Journal_Article[contains-text-basic(@Title, " +Java -XML +'JDK 1.3'")=1] 
SORTBY(score-basic(@Title, " +Java -XML +'JDK 1.3' "))
```

This query finds all journal articles that contain the text “Java” and the text “JDK 1.3” but not the text “XML” using the simplified (basic) text search syntax and sort the results by the text search score.

**Advanced text search**

Advanced text search syntax is used to allow the user to specify more complex conditions for text search. The text search uses the NSE text search syntax, and allows such powerful features as proximity search and fuzzy search. Advanced text search syntax uses contains-text and score functions similar to the way the contains-text-basic and score-basic functions are used for the basic text search.

The strings that are supplied to the advanced functions should be in NSE syntax, except as follows: change double quotes to single quotes, and vice-versa. For example, CONTAINS (description," 'IBM' ")=1 condition in NSE becomes contains-text(@description," 'IBM' ")=1 in CM query language. This needs to be done to support simplicity of writing queries with minimal use of escape characters. To check if it is true, equate the contains-text function to 1; to check if it is false, equate the contains-text function to 0.

We use several examples to demonstrate this:

The following example shows an advanced text search using contains-text and score functions:

```
//Journal_Article[Journal_Author/@LastName = "Smith" AND 
contains-text(@ArticleText, " 'Java' & 'XML' ")=1] 
SORTBY(score(@ArticleText, " 'Java' & 'XML' "))
```
This query finds journal articles with author Smith that contain the text “Java” and the text “XML”. The results are ordered by the text search score. For the syntax supported by this function, see the NSE documentation. Note that the contains-text function should be equated with 1 to be true and 0 to be false. The score function uses the ranking information returned by NSE, which is used in this case to sort the resulting journal articles through SORTBY.

- The next example of advanced text search uses contains-text and attribute sorting.

```xml
/Journal[Journal_Article[contains-text(@Title, " 'CM' | 'eClient' ")=1]]
SORTBY (@Title DESCENDING)
```

This query finds all journals that have either the word CM or the word eClient in their titles and sorts the results in descending order by their titles. The sorting in this case uses the DESCENDING operator on the Title attribute. The default for the SORTBY is ASCENDING.

### 7.3.4 Links

The Content Manager query language enables you to traverse links. We use several examples to demonstrate this:

- The following example demonstrates how to do link traversal:

```xml
/SIG[@Title = "SIGMOD"]/OUTBOUNDLINK[@LINKTYPE = "contains"]/TARGETITEMREF => Journal[Journal_Editor/@LastName = "Smith"]/Journal_Article
```

This query finds all articles in journals edited by Smith that are contained in SIGs with title “SIGMOD”. It is an example of following links in the forward direction. The component OUTBOUNDLINK and its attribute TARGETITEMREF are used to traverse to all Journals and then finally the underlying Journal_Articles. The last component in the path is what is returned as the result of the query. The result can be constrained by traversing only specific link types (contains in this example) to a specific type of items (Journal in this example). Since, at the conceptual level, inbound and outbound links are looked at as being parts of items, the de-referencing operator can be used to relieve applications from writing explicit joins.

- The next example follows links in the backward direction:

```xml
/Journal[@Cost < 5 AND .//Journal_Author/@LastName = "Smith"]
/INBOUNDLINK[@LINKTYPE = "contains"]/SOURCEITEMREF => *
```

This query finds all items of any type that have journals which cost less than five dollars with articles by author Smith. The wildcard “*”, following the de-referencing operator “=>” ensures that the items of ANY type are returned as the result.
7.3.5 References

The Content Manager query language also enables you to traverse references in either direction. We use several examples to demonstrate this:

- The following example demonstrates traverse references:

  ```xml
  /Conference/Conference_Note[@PublicationRef => Book[@Title LIKE "%eClient%"]]/Conference_FAQ
  ```

  This query finds all the frequently asked questions for conferences, for which the conference notes refer to books with titles mentioning eClient. Note that PublicationRef is a reference attribute and that the de-reference symbol “=>” is used to follow that reference.

- The following example also demonstrates traverse references:

  ```xml
  /Conference[@Title LIKE "%eClient%"]/Conference_Note/@PublicationRef => /Book_Chapter
  ```

  This query finds all chapters of books referenced in the notes of conferences related to eClient. It contains traverse references in forward direction.

- The next example uses traverse references in reverse direction:

  ```xml
  /Book/REFERENCEDBY/@REFERENCER => *
  ```

  This query finds all the components that have references pointing to any books. Take note of the component REFERENCEDBY and its attribute REFERENCER that are used to do this reverse traversal.

- The next example also uses traverse references in reverse direction:

  ```xml
  /Book[@Title LIKE "XML"]/REFERENCEDBY/@REFERENCER => Conference_Note/Conference_FAQ
  ```

  This query finds all the frequently asked questions under conference notes that refer to books about XML. Note that since the reference attributes originate inside of the Conference_Note component, this is the component that must appear as the first component after the de-reference operator. This query produces an empty result set if, for example, Conference follows the “=>” operator.

- The following query is yet another example of traverse references in reverse direction:

  ```xml
  /Book/REFERENCEDBY/@REFERENCER => *[@Remark LIKE "%XML%"
  ```

  This query finds all the components that contain XML in their remarks and that have references pointing to books.
7.3.6 Versions

By default, all versions of a matching item are returned when performing a search. We use several examples to demonstrate version searching:

- The following example shows how to use search using the latest-version function:
  
  \[
  /\text{Journal}[@\text{VERSIONID} = \text{latest-version(.)}]
  \]

  This query finds all the journals of the latest version. VERSIONID is a system-defined attribute that is contained in every component type and can be used to specify a particular version.

- The following query is another example that shows how to use the latest-version function on the target of a de-reference:
  
  \[
  /\text{Conference/Conference Note}@\text{SYSREFERENCEATTRA} => /\text{Book}[@\text{VERSIONID} = \text{latest-version(.)}]
  \]

  This query finds all the books of the latest version that are referenced in the notes of any conferences.

- The following query is an example that shows how to use the latest-version function on wildcard components:
  
  \[
  /\text{Book/RE}F\text{ERENCEDBY/REFERENCE}R => * [@\text{VERSIONID} = \text{latest-version(.)}]
  \]

  This query finds all the components of the latest version of any item that have references pointing to any books.

7.3.7 System-defined attributes

You can query against attributes defined by the system, as shown in the following example:

\[
/*[@\text{ITEMID} = "A1001001A01J09B00241C95000"]
\]

This query finds all the root components with a specific item ID.

7.3.8 Resource items

You can query against the attributes of a resource item, or directly against the resource itself. The following example uses advanced text search on resource items:

\[
/\text{TextResource[contains-text(@TIEREF, " 'Java' & 'XML'")=1}]
\]

This query finds text resources in a text resource item type TextResource that contain the text Java and the text XML. Note that the TIEREF attribute is used as a representation of the resource represented by the item of type TextResource. NSE syntax is used as usual in this case inside the contains-text function.
7.3.9 Document parts

The Content Manager query language enables you to fully interrogate the document model. We demonstrate this with several examples:

- The following example performs text search on a document model:
  
  \[
  \text{Doc[contains-text(.//ICMPARTS/@TIEREF, ''XML'')=1]}
  \]

  This query finds all documents that contain the word XML in any one of its parts. The query language offers a virtual component ICMPARTS that allows access to all the ICM Parts item types contained under a specific item type of Document classification.

- The following example access to ICMPARTS:
  
  \[
  \text{Doc[@ArchiveID = 555]/ICMPARTS/@SYSREFERENCEATTRS => *}
  \]

  This query finds all the parts of the document with the storage ID of 555.

- The following example also access to ICMPARTS:
  
  \[
  //ICMPARTS/@SYSREFERENCEATTRS => *
  \]

  This query finds all the parts in all of the documents in the system. Because both the Doc and Paper item types have been defined as being Documents in the system, the ICM Parts from both of them are returned in the result.

7.3.10 Lists

The Content Manager query language has the ability to deal with lists in many contexts. We provide several examples:

- The following example works with a list of literals and expressions:
  
  \[
  \text{Journal[@Title = [Journal_Article/@Title, .//Journal_Section/@Title, "IBM Systems Journal"]}}
  \]

  This query finds all journals that have a title that is equal to either its article’s title, its section’s title, or “IBM Systems Journal”.

- The following example works with a list of numeric literals:
  
  \[
  \text{Book[@Cost = [10, 20, 30]]}
  \]

  This query finds all books that cost either $10, or $20, or $30.

- The following example works with a list of the results from a query:
  
  \[
  \text{[Journal, /Book[@Title = "CM"]]}\]

  This query finds all journals or all books with the title CM.
7.3.11 Attribute groups

Text search on attributes within an attribute group is slightly different, in that any references to the attributes in a group must include the group name.

The following example demonstrates this:

```
/Doc[Doc_Description/@PageSummary.NumPages >= 20]//Doc_Details
```

This query finds all the details on the documents in which the description is at least 20 pages long. Note that if an attribute (for example, NumPages) is contained within an attribute group (for example, PageSummary), then you must refer to that attribute as GroupName.AttrName (for example, PageSummary.NumPages). The attribute @NumPages would not be found under Doc_Description.

7.3.12 Set operators

To make more advanced queries, you can use the UNION, INTERSECT and EXCEPT operators.

**Note:** The intermediate results obtained by INTERSECT/EXCEPT cannot be combined with arithmetic (unary/binary) or comparison operators. They can be combined by set operators (UNION/INTERSECT/EXCEPT) or appear by themselves.

The following examples demonstrate the usage:

- **This is an example using EXCEPT operator:**

  ```
  (/Journal/Journal_Article[@Title = "CM"] EXCEPT //Journal_Article[@Classification = "Security"])//Journal_Section
  ```

- **This is an example using UNION operator:**

  ```
  /Journal[(Journal_Editor/@LastName UNION .//Journal_Author/@LastName) = "Smith"]
  ```

- **This is an example using INTERSECT operator:**

  ```
  /Journal[Journal_Article[Journal_Section/@Title INTERSECT .//Journal_Figure/@Caption]/@Title = "CM"]
  ```

- **This is another example using INTERSECT operator:**

  ```
  /Journal[@Title = "CM"] UNION /Journal[@Cost = 20] INTERSECT /Journal[@Organization = "IBM"]
  ```
7.3.13 Row-based view filtering

With row-based view filtering, you can filter a component based on the contents of one of the component's attributes. By having different views on the same item type with different filtering conditions, you can separate the data for an item type into logical blocks, allowing users to view only certain data, depending on which view is used to access the data. Therefore, in query, row-based view filtering helps to automatically limit the amount of data retrieved for a given view.

**Note:** Improperly using row-based view filtering can result in a significant increase in the length of the generated SQL and a decrease in query performance. In the DB2 Content Manager system, your query gets converted to a SQL query string that is executed on the underlying database tables. Since database systems have a limit on the length of the SQL query string, improper usage of filtering can cause this string to become so long that it can exceed the limit and prevent successful execution of your query. You should review the performance discussion before you decide to use this feature. Also, when defining row-based filters, be sure to create database indexes against the attributes you are using in the filter. This will improve the performance of your query.

We provide a simple scenario that describes, from a high-level perspective, how row-based view filtering can be used in the DB2 Content Manager system. These are the steps involved in the scenario:

1. Define an item type called Journal.
2. Add some items to this item type.
3. Define an item type view called MyJournal with the following filter:
   ```
   @Organization = "IBM"
   ```
4. Execute a query against the item type view MyJournal.
5. Display the results of the query to the user. Only journals for the IBM organization are returned.

The following examples demonstrate the usage:

- There are 1,000,000 components of the component type Journal in the system. 1,000 of these components have IBM in the Organization attribute. You execute the following query to get all journals associated with the view MyJournal (in the example: 1,000 results):
  ```
  /MyJournal
  ```
- This is an example where we search for all Journals where Organization = IBM and Title starts with CM.
  ```
  /MyJournal[(@Title LIKE "CM%")]
  ```
7.3.14 Query on checked-out items

Using the ICMCHECKEDOUT element allows you to search for items in Content Manager V8.3 that have been checked out. This element is a sub-element of only the root components, but not of the descendant components.

Whenever an item is checked out, all versions of that item are checked out. Therefore, when an ICMCHECKEDOUT element is applied to a checked out item, all currently available versions will be returned. To retrieve a specific version, you can still use the @VERSIONID query syntax.

The following examples demonstrate the usage:

- This query finds all Journals checked out by SMITH:
  /Journal [ICMCHECKEDOUT]

  **Note:** The value for ICMCHKOUTUSER must be entered in *upper case* in a query. Since the content servers store user IDs as upper case, all queries must query for user IDs using upper cases. All attribute data pertaining to user IDs must store them in upper case as well.

- This query finds the latest version of all Journals that have been checked out:
  /Journal [ICMCHECKEDOUT AND @VERSIONID = latest-version()]

- This query finds all Journals checked out after 2003-08-02-17.29.23.977001:
  Journal [ICMCHECKEDOUT/@ICMCHKOUTTS > "2003-08-02-17.29.23.977001"]

The reserved elements that can be used include: ICMCHECKEDOUT, ICMCHKOUTUSER, ICMCHKOUTTS.

7.4 Using query language

To perform a search, you need to develop an application using the Content Manager APIs. There are three things to consider when implementing searches:

- Query string
- Query options
- Query results

7.4.1 Query string

The query string defines which item types and what search criteria to use. These are discussed in detail in 7.3, “Query strings” on page 172.
7.4.2 Query options

Query options are available to specify parameters such as the maximum number of results, prefetch size, and content retrieve scope. The following constants are defined in com.ibm.mm.sdk.common.DKConstant:

- **DK_CM_PARM_MAX_RESULTS**: Specifies the maximum number of results to return in a search. Set to 0 for no maximum.
- **DK_CM_PREPARE_QUERY**: Prepares the query. Does not open the Result Set Cursor. The application must explicitly open the cursor and execute the query at a later time. The value specified is ignored.
- **DK_CM_PARM_PREFETCH_SIZE**: The block size in which results are retrieved from the datastore (if not using default). A small value allows for faster display of the first $n$ results, but may increases overall retrieval time for the entire result set.
- **DK_CM_PARM_RETRIEVE**: Retrieval options to apply for retrieving each result.
- **DK_CM_PARM_END**: Marks the end of the set of search options (see Example 7-1 for usage).

Note: You create an options array and insert a DKNVPair into each element of the array for each search option you want to include.

**Tip**: For more information on Retrieval options, please refer to SItemRetrievalICM in the sample code provided with Content Manager.

Example 7-1 shows how to specify search and query options.

**Example 7-1  Specifying search / query options**

```java
// Specify Search / Query Options
DKNVPair options[] = new DKNVPair[3];
options[0] = new DKNVPair(DK_CM_PARM_MAX_RESULTS, "50");
options[1] = new DKNVPair(DKConstant.DK_CM_PARM_RETRIEVE,
                           new Integer(DKConstant.DK_CM_CONTENT_ATTRONLY));

// Must mark the end of the NVPair
options[2] = new DKNVPair(DKConstant.DK_CM_PARM_END, null);
```

Important: You must mark the end of the DKNVPair array by placing a DKNVPair of type DKConstant.DK_CM_PARM_END into the last element of the array.
7.4.3 Query results

When executing a query, you have three choices for how and when searches should be executed and results should be returned:

- **Execute**: Returns a cursor that may be used to iterate over a collection of results. Each result is retrieved in blocks, managed by the system as the application explicitly iterates through. This is especially beneficial for large numbers of results. The application is responsible for explicitly determining when to spend the time retrieving blocks of results.

- **Evaluate**: Returns all results as a collection. All items are retrieved at once during this operation. This is especially beneficial for a small number of results. A large number of results may require a long time since all results must be retrieved at once, if a maximum limit option is not specified. If there is no need to retrieve all results immediately and the result set is large, execute is the best alternative.

- **Evaluate with callback**: Spawns a separate thread to execute the query and retrieve results. The thread iterates over the result set, calling a callback object for each block of results. A callback object is any object that implements the dkCallback interface. This object is defined and created by the application.

Example 7-2 demonstrates how you can implement a query. It does the following tasks:

1. Create a datastore.
2. Define query options (maximum results is 5).
3. Establish a connection to the datastore.
4. Create a query string to find all journals with Java in their title.
5. Execute the query.
6. Process the results from the query using a cursor.

**Example 7-2 Implementing a Content Manager query**

```java
// Create a Datastore
DKDatastoreICM dsICM = new DKDatastoreICM();

// Specify max results for query
DKNVPair parms[] = new DKNVPair[2];
String strMax = "5";
parms[0] = new DKNVPair(DK_CM_PARM_MAX_RESULTS,strMax);
parms[1] = new DKNVPair(DK_CM_PARM_END,null);

// Establish a connection
dsICM.connect("ICMNLSDB", "ICMADMIN", "PASSWORD");
```
// Create ICM query expression
String qs = "/Journal[like(@Title, "Java\%")];

// Execute a parametric query
dkResultSetCursor rsc = dsICM.execute(qs, DK_CM_XQPE_QL_TYPE, parms);

// process results in cursor
DKDDO ddo = (DKDDO) rsc.fetchNext(); // fetch an item from the cursor

7.5 SQL queries

Since Content Manager data is all stored in an underlying Relational Database Manager System (RDBMS), it is possible to run SQL queries directly against the database server, rather than going through the Content Manager query language and API. However, the database schema is complicated, and we recommend it for advanced users only. We recommend that you use SQL queries only to view data in Content Manager tables and never to modify or delete data. The database schema is documented in the Content Manager V8.3 Information Center, found in the section titled API Reference.

IBM DB2 Content Manager V8.3 Information Center can be found from the following URL:

http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0

7.6 Other resources

There are many good resources to use for further information on the Content Manager query language. *IBM DB2 Content Manager Enterprise Edition V8.3: Application Programming Guide*, SC18-9679, has in-depth information on searching a Content Manager system.

In addition, the sample code provided with the installation of Information Integrator for Content, has excellent explanations and sample code discussing and demonstrating the query language. We recommend the following samples:

- SSearchICM: This sample discusses the query components and syntax and demonstrates how to implement a query in your own application.
- SSearchCallbackObjectICM: This sample demonstrates how to implement a callback object that is used when using the evaluate with callback option to process query results.
Security

In this chapter, we provide an overview of the security options available to protect data stored within a Content Manager system. It includes a section on the user exits available and what they can be used for. It covers both authenticating users and authorizing actions that these users can perform.

There are also sections explaining how to perform simple security related operations on a Content Manager system, such as changing passwords.

In the last section of this chapter, we also provide information on how to integrate a host security system such as RACF®.
8.1 Content Manager security concepts

Security is engineered into Content Manager to give you confidence that your data is accessed by authorized users only, and to make sure that once these authorized users have access to your system, they can only perform actions on the data that you have defined in advance for them.

Content Manager V8 provides a complete, integrated, extensible and secure control environment for managing and processing content, while leveraging industry standards such as LDAP (Lightweight Directory Access Protocol).

With Content Manager you can choose anything from a flexible and open system with client authentication within a trusted environment, to a more rigid security model where every user must be authenticated by the server.

8.1.1 Content Manager security

The Library Server manages the relationships between items in the system and controls access to all of the system information, including the information stored in the Resource Managers configured in the system. The Library Server controls access to objects.

Content Manager can secure items based on access control lists. These apply to folders, documents and their component parts, which are versions, notes and annotations. Sections of documents such as worksheets form part of the entire document and are stored in their entirety; access control can only be applied at the document level.

All passwords that pass between Content Manager clients and servers are encrypted or encoded. As a further security measure, the System Administration Client communicates with the Resource Manager through the Secure Sockets Layer (SSL) interface. It should be noted that there is no logging of attempted and failed logons. The number of failed attempts to logon can be set within the Library Server configuration, by using the Content Manager System Administration Client.

Note: On z/OS there is no communication between the Content Manager administration client and the Resource Manager.

Users store and retrieve objects to and from the Resource Manager by issuing requests through the Library Server. When a request is granted, the Library Server returns a security token and the location of the object to the users. When retrieving content, the client uses the security token to access the Resource Manager and gives the location of the object to find the object. The object is then returned to the client and copied to the staging area.
Objects in the V8 Resource Manager per default are stored in the file system, as they were for the Content Manager V7 Object Server. It is important to secure your server environment to prevent unauthorized access. This is done at the operating system level and not at the application level. No encryption of objects takes place within Content Manager.

8.2 Authentication

Authentication is the method by which a system identifies if a user is who they claim to be, via the use of user IDs and passwords, and whether this user is defined within the system.

Content Manager server authentication
User IDs and passwords can be managed internally by Content Manager, eliminating the need to define users to other security systems. In this instance the Library Server manages and stores user information, which is used to authenticate users during logon requests.

LDAP authentication
If a requirement exists to manage Content Manager user IDs and passwords at an enterprise level, rather than on a system-by-system basis, LDAP integration can be exploited.

Content Manager supports importing of users and user authentication using the standard LDAP protocols, from:

- IBM Tivoli Directory Server (previously known as IBM SecureWay® Directory)
- Lotus Domino Directory Notes Address Book (NAB)
- Sun Java™ System Directory Server
- eDirectory (Novell)

Administration utilities are provided to import LDAP users, as well as to synchronize Content Manager with an LDAP directory. By utilizing LDAP, Content Manager users can be managed centrally within an organization, which can help to reduce administrative overheads that occur when attempting to manage numerous disbursed pools of users simultaneously.

In addition, Content Manager provides a server side security exit that can be used to integrate with the above LDAP servers. The support of LDAP technologies enables WebSphere single sign-on capabilities for Content Manager.
8.2.1 LDAP integration overview

During the Content Manager installation, you decide if you are going to use the standard method for managing users, or if you are going to use Lightweight Directory Access Protocol (LDAP). You can decide to enable LDAP at that time, or you can decide to enable it later by using the System Administration Client.

When a user logs on to Content Manager, the user exit is called and authentication is performed on the LDAP server. The user ID resides on both the Library Server and the LDAP server. The user password resides only on the LDAP server.

To enable LDAP:
1. Launch the System Administration Client.
2. Bring up the LDAP Configuration window by selecting Tools → LDAP Configuration.
3. Select the Enable LDAP User import and authentication check box (see Figure 8-1).
4. Provide the LDAP server information on the Server page.
5. Click OK.
After you enable LDAP, you can import users by clicking the LDAP button in the New User window. This allows the users from the LDAP server to be selectively imported into Content Manager. Alternatively, you can import users in groups using the LDAP User ID Import Scheduler utility. During logon, the Library Server calls the user exit that connects to the LDAP server to authenticate the user. If the LDAP server is not able to verify the user and the password, the logon process might terminate depending on the reason code returned from the user exit.

You can modify the LDAP server configurations after enabling LDAP by going to the main Content Manager System Administration Client window, and select Tools → LDAP Configuration.

You can also change your current LDAP server by going to the LDAP User Registry Import Utility from the Start → Programs → IBM Content Manager for Multiplatforms → System Administration Client.

**Figure 8-1 Enabling LDAP user import and authentication**
8.2.2 Single sign-on

With single sign-on, users can log on once to either a Web site or desktop system and not have to log on to different applications from the same Web site or desktop system. Content Manager provides two types of single sign-on capabilities for two environments, Web and desktop:

- Single sign-on through WebSphere security
- Single sign-on through workstation authentication

**Single sign-on through WebSphere security**

You can use this function with Web applications in a WebSphere Application Server environment to take advantage of WebSphere security and its single sign-on capability.

**Note:** In order to enable single sign-on through WebSphere Application Server, WebSphere global security (see 8.5, “WebSphere global security” on page 223) must be enabled with Light Weight Third-party Authentication (LTPA) specified as the active authentication mechanism.

Figure 8-2 illustrates the basic elements of the authentication and evaluation process for WebSphere Application Server. Netscape Navigator's approach to single sign-on replaces client authentication based on passwords sent over the network, with client authentication based on the Secure Sockets Layer (SSL) and Certificates.
To enable this authentication mechanism, you need to set up the Library Server in a trusted logon mode by completing the following setup steps using the Content Manager System Administration Client:

1. Go to the Library Server Configuration window within the Content Manager System Administration Client and check **Allow trusted logon** (Figure 8-3).
2. Include the privilege AllowTrustedLogon in the Privilege Set of the Content Manager user ID that you want to allow for single sign-on.

3. The shared database connection ID must have the UserDB2TrustedConnect privilege set. To accomplish this, in the System Administration Client, select Tools → Manage Database Connection ID → Change Shared Database Connection ID. Provide the password for the shared connection ID, and then make sure the check box labeled Password is required for all users is unchecked before saving the change.

**Note:** Web applications must use the connectWithCredential() method instead of the connect() method. Users must be imported from LDAP to Content Manager.

**Single sign-on through workstation authentication**

Single sign-on through workstation authentication is also known as Unified logon. When you set up your Library Server for this Unified logon, you let users have access to Content Manager using their workstation ID and password without prompting for an additional user ID and password.
To enable this single sign-on feature, use the following steps:

1. Enable the server for single sign-on during the installation as follows:
   - Check the **Enable single sign on** check box option.
   - For authentication type, select the **Client** (instead of server) option.

   If you did not enable it during installation, you can use the tool we provided to update the cmbicmsrvs.ini file. Refer to the *Content Manager Planning and Installation Guide* for more information on the tool that works with this INI file.

2. On DB2 server side, set the database manager authentication to CLIENT from the DB2 database manager configuration.

3. Define the workstation logon user ID in Content Manager.

### 8.3 Authorization

Authorization is the process of establishing whether a user has the permission (or privilege) necessary to perform the requested action upon an entity.

The Content Manager access control model is comprised of the following fundamental elements:

- Privileges and privilege sets
- Controlled entities
- Users and user groups
- Access control lists

The various access control elements work as follows. Each Content Manager user is granted a set of user privileges. These privileges define the operation a user can perform. A user's effective access rights never exceed the user's defined privileges.

The access control model of Content Manager is applied to the controlled entity. A *controlled entity* is a unit of protected user data. In Content Manager, the access control for a controlled entity can be at the item level or item type level. For example, you can bind an ACL to an item type to enforce access control at the item type level. You can also bind an ACL at an item level to enforce access control at individual items. Operations on controlled entities are regulated by one or more control rules, called access control lists (ACLs). Every controlled entity in Content Manager system must be bound to an ACL.

When a user initiates an operation on an item, the system checks the user's privilege and the ACL bound to the item to determine if the user has the right to do such an operation on the item.
Figure 8-4 shows an example of how the system determines user's access rights to an item, based on privileges and ACLs.

### 8.3.1 Privileges

Privileges grant the right to perform a specific action on a specific item in the system, such as create an item or delete one. Every Content Manager user is granted a set of user privileges. The privileges define the maximum operations a user can perform on information in the Content Manager system. A user's access rights do not exceed the defined user privileges for the user.

Your first task in managing access is to create privilege sets for users. A privilege set identifies the tasks or actions that a user can perform. Privilege sets combine privileges and are tailored for certain types of users.

Content Manager provides a number of pre-defined privileges that you cannot change, called system-defined privileges, that you can group together to create a privilege set. You can also define your own privileges (these can only be used within a custom application), called user-defined privileges. You enforce user-defined privileges in your application using user exit routines. You then assign the privilege sets that you create to individual users. You cannot assign a privilege set to a user group.
The Content Manager administration client provides privilege groups, privilege sets, and individual privileges, defined as follows:

**Privilege**
A privilege represents a user action.

**Privilege group**
A privilege group is a collection of user tasks for the purpose of helping administrators create new privilege sets or user roles in the privilege set dialog.

**Privilege sets**
A group of privileges assigned to a user is a privilege set. For example, one privilege set can contain the privileges create, update, and delete. Privilege sets allow for easier system administration. You must group privileges into a set before you can use them. There is no limitation on the number of privileges a set can contain.

Every privilege has a system-generated, unique code called a privilege definition code. The privilege definition codes from 0 to 999 are system-defined privileges. User-defined privileges start from 1000 and above.

The system-defined privileges are classified into two categories: system administration privileges, and data access privileges. You can use the system administration privileges to model user data and administer and maintain the Content Manager system. You need system administration privileges to complete tasks such as configuring the system, managing the Library Server configuration, and managing item types. You can use the data access privileges to access and change the system data, like items and item types.

Some of the Content Manager pre-defined privilege sets are:

**AllPrivs**
Users with this privilege set can perform all functions on all Content Manager entities.

**NoPrivs**
Users with this privilege set cannot perform any functions on any Content Manager entities.

**SysAdminCM**
Users with this privilege set can perform all Content Manager system administration and data modeling functions.

**Creating a new privilege set**
There are two ways that you can create a new privilege set:

- Creating a privilege set (basic)
- Creating a privilege set (advanced)

In the basic creation, you can create a privilege set by selecting roles that have a description of what actions the role will allow (this is new in Content Manager V8.3). In the advanced creation, you can create all aspects of the privilege set, including individual privileges.
Table 8-1 shows some of the database tables used by Content Manager to define privileges, privilege groups, privilege sets and their behavior:

<table>
<thead>
<tr>
<th>Database table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMSTPRIVDEFS</td>
<td>Contains listing of privilege codes (corresponding privilege name is in ICMSTNLSKEYWORDS table - KEYWORDCLASS=10)</td>
</tr>
<tr>
<td>ICMSTPRIVGROUPCODE</td>
<td>Contains listing of privilege group codes (corresponding privilege group name is in ICMSTNLSKEYWORDS table - KEYWORDCLASS=12)</td>
</tr>
<tr>
<td>ICMSTPRIVSETCODES</td>
<td>Contains listing of privilege set codes (corresponding privilege set name is in ICMSTNLSKEYWORDS - KEYWORDCLASS=11)</td>
</tr>
<tr>
<td>ICMSTPRIVSETS</td>
<td>Contains the relationship between privileges and privilege sets</td>
</tr>
<tr>
<td>ICMSTDOMAINPRIVSET</td>
<td>Contains the relationship between privilege sets and domains</td>
</tr>
<tr>
<td>ICMSTCOMPILEDPERM</td>
<td>Contains the general privileges for users</td>
</tr>
</tbody>
</table>

### 8.3.2 Users and user groups

In most situations there are groups of users that require the same type of access to the system. For example, all of the editors in a publishing company require search, retrieve, and update privileges to the articles item type. You can group the editors and any other users with common access needs into a user group. You cannot, however, put one user group into another user group.
A user group is solely an aid to convenience by grouping individual users who perform similar tasks together. A user group consists of zero or more users. You do not assign a user group a privilege set. Each user in a user group has a privilege set. A user group makes it easier to create access control lists for objects in your system.

By default, there is one Content Manager user group created when the system is first installed. This is the ICMPUBLIC group, which is a special Content Manager user group, to which every Content Manager user belongs.

If you have domains enabled (8.3.9, “Domains” on page 214), before you assign a user ID to a group, check to see if that user group is in a specific domain or the PUBLIC domain. Make sure that the user group is in the domain that you want your user ID to be in. If you want to create a user ID specifically for a domain, you can click New User within the User Group window. You can then add the user that you create to the user group, and ensure that the user is in the same domain.

### Assigning users to Resource Managers

To allow users to access a specific Resource Manager, you assign a Resource Manager to a domain that users have access to.

### Assigning users to collections

To allow users to access to collections, you assign a collection on a Resource Manager to a domain that users have access to.

### Assigning users to a default ACL

Depending on the default ACL choice, the ACL for the user might get assigned to the items when an ACL is not given at the time the item is created. The default ACL choice determines whether the item type’s ACL or the user’s ACL gets assigned. The Library Server checks for the ACL for each item when the ACL binding is set at item level binding.

Table 8-2 shows which database tables hold the definitions for Content Manager users and groups:

<table>
<thead>
<tr>
<th>Database table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMSTUSERS</td>
<td>User and user group definition table (USERKIND=0 for users USERKIND=1 for groups)</td>
</tr>
<tr>
<td>ICMSTUSERGROUPS</td>
<td>Relationships between users and user groups</td>
</tr>
</tbody>
</table>
### 8.3.3 Creating user IDs and passwords

If you want a user ID that you define in the System Administration Client to also be used for DB2 authentication, then the user ID must follow the DB2 naming rules. The DB2 naming rules apply for user IDs that you want to use for either super administrators or connect user IDs. You cannot use the following words:

- USERS
- ADMINS
- GUESTS
- PUBLIC
- LOCAL
- Any SQL reserved word listed in the SQL Reference.

You cannot begin a user ID with the following characters:

- SQL
- SYS
- IBM

You can use the following characters:

- A through Z Restriction: some operating systems allow case-sensitive user IDs and passwords. Check your operating system documentation to see if it allows for case-sensitivity.
- 0 through 9
- #
- $

**Attention:** Content Manager user IDs cannot exceed 32 characters.

When creating a user, the following information is required (see Figure 8-5):

- Name and password:
  The password is not required when you are creating an administration user. We recommend not to specify the password for administration users.
- Maximum privilege set for the user
- Default Resource Manager and collection
- Default item ACL
- User group (optional)
- Admin domain:
  - Required when domains are enabled
  - Only one domain can be specified
  - Cannot be the public domain
8.3.4 DB2 administration authority

When logging on to the System Administration Client, there are two levels of authentication: one at the database level and another at the product level. Administrators have two classifications when you enable the administrative domains feature (see 8.3.9, “Domains” on page 214): super administrators and sub-administrators. In general, only super administrators have access to the System Administration Client.

Super administrators must have DB2 administration privileges. This user ID has to be defined in the operating system with the db2admin privilege. The password for this operating system ID is used to connect to DB2 and to log on to the Library Server. The password defined for the Library Server is not used. This user ID is defined in the Library Server with full Content Manager administration privileges (AllPrivs privilege set) to do all administration activities.
Sub-administrators do not require DB2 privileges. They manage only certain objects of the Library Server. They log on to the System Administration Client one of two ways:

- If the user ID is an operating system user ID, then the password in the operating system is used to connect to DB2 and to log on to the Library Server.

- If the user ID is not an operating system user ID, then the shared database connection user ID and password pair encrypted in the cmbfedenv.ini file (for Information Intergrater for Content) or the cmbicmenv.ini file (for Content Manager) is used to connect to DB2, and the user ID and password provided in the Logon window is used to log on to the Library Server.

Sub-administrators also need the Content Manager privileges. They need the Domain Administrative privilege for all sub-domain administration activities.

**Connecting to DB2 using the INI files**

Each entry in the INI file contains the name of a Library Server and a pair of encrypted user ID and password for connecting to DB2. This encrypted user ID (known as connect user ID) and password are defined at the time you install the product. The connect user ID must be different than the system administrator’s user ID. Content Manager uses cmbicmenv.ini for connecting to DB2. The default connect user ID is ICMCONCT.

During installation, the passwords for the Library Server and the Resource Manager are contained in three places: The cmbicmenv.ini file contains the user ID and password to access the Library Server. The operating system defines access to the database where the Library Server and Resource Manager reside. The ICMRM.properties file contains the Resource Manager user ID and password.

If the INI file is used, that is, the user ID is not an operating system user ID, then both the user ID and the connect user ID in the INI file must exist in the Library Server.
The connect user ID must be defined in the Library Server and operating system. It does require the UserDB2Connect privilege. To change the connect user ID and password in the INI file, select **Tools → Manage Database Connection ID → Change Shared Database Connection ID** from the System Administration Client window. See Figure 8-6.

![Change Shared Database Connection ID and Password](image)

**Figure 8-6  Updating Library Server connect user ID**

### 8.3.5 Changing password to Resource Manager

If you need to change the password to the Resource Manager, then you need to change the password for the logon of the Library Server to the Resource Manager and the system administrator’s password to the Resource Manager.

**Important:** When changing the passwords for the logon of the Library Server and system administrator to the Resource Manager, complete the following steps in order:

1. Log on to the Content Manager System Administration Client.
2. Expand the Resource Manager tree.
3. Click the Resource Manager that you want to modify and expand its tree.
4. Click **Server Definitions**, select your Resource Manager in the right-hand window, and select Properties. The **Server Definition Properties** window opens (see Figure 8-7).

![Server Definition Properties - RMDB](image)

**Figure 8-7 Resource Manager server definitions properties window**

5. Change the password in the Password field.

6. Click **OK**.

7. Right-click the Resource Manager that you expanded (from step 2) and select **Properties**. The Resource Manager Properties window opens (see Figure 8-8).
8. Change the password in the Password field and click OK.

8.3.6 Changing database access passwords

To change the database access passwords, you need to change the operating system password for the database connection and the ICMRM.properties file so that the resource manager can identify the new password.

To change the operating system password for the database connection, perform the following steps:

1. Depending on your operating system, navigate to the Users and Passwords utility.
2. Click RMADMIN.
3. Select Set Password.
4. Enter the new password.
To change the ICMRM.properties file, complete the following steps:

1. Open the ICMRM.properties file. The default location is:
   
   C:\WebSphere\AppServer\installedApps\<hostname>\icmrm.ear\icmrm.war\WEB-INF\classes\com\ibm\mm\icmrm\ICMRM.properties for Windows, for AIX replace C:\WebSphere with /usr/WebSphere and for Solaris replace C:\WebSphere with /opt/WebSphere.

2. Change the DBPassword to match the operating system password.

3. Save the ICMRM.properties file.

After you change the database password, the database needs to either be restarted, or, you can let it issue two or three errors until it resets itself.

### 8.3.7 Access control lists

An access control list (ACL) is a list consisting of one or more individual user IDs or user groups and their associated privileges. You use ACLs to control user access to objects in the Content Manager system. The objects that can be identified in ACLs are:

- Objects stored by users
- Item types
- Item type subsets/views
- Document Parts
- Workbaskets
- Collection Points
- Business Applications
- Processes
- Worklists

Although privilege sets define an individual user’s maximum ability to use the system, an ACL restricts an individual user’s access to an object. When a user takes action on an item controlled by an access control list, the system compares the privileges assigned to the user in the user’s assigned privilege set to the privileges granted the user in the ACL controlling the item.

Even though the ACL may grant the user additional privileges beyond their user profile defined privileges, the user can only perform actions where there are matches between the user profile privileges and the ACL privileges granted the user. An ACL that has a privilege that is not defined by a user’s privilege set does not grant the user that privilege. Only users who have that privilege can use that privilege on an object, as demonstrated by Example 8-1. An ACL limits user access; it does not grant more access. ACLs provide another level of security when managing a system.
Example 8-1  ACLs and privileges working together

If the ACL of an item allows it to be deleted, but the user attempting to delete the item does not have the delete privilege, then the user cannot delete the item.

Table 8-3 shows some of the Content Manager database tables related to ACLs and their function.

Table 8-3  Library Server database tables related to ACLs

<table>
<thead>
<tr>
<th>Database table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMSTACCESSCODES</td>
<td>Contains a list of the ACL codes (corresponding ACLCode name is in ICMSTNLSKEYWORDS table - KEYWORDCLASS=13)</td>
</tr>
<tr>
<td>ICMSTDOMAINACCESS</td>
<td>Contains the relationship between the ACL code and domains</td>
</tr>
<tr>
<td>ICMSTACCESSLISTS</td>
<td>Contains the ACL rules</td>
</tr>
<tr>
<td>ICMSTCOMPILEDACL</td>
<td>Contains a user privileges, based on the intersection between a users general privileges and the ACL rules (this pre-computed access list on the Library Server greatly increases privilege processing performance)</td>
</tr>
</tbody>
</table>

A controlled entity is bound to a specific ACL through the ACL code. When associated with controlled entities, ACLs define the authorization of the bound entities and do not circumvent the user privileges. An ACL is enforced, and user privileges are checked. When the privilege ItemSuperAccess is assigned to a user, the user is allowed to bypass ACL checking.

The users specified in access control rules can be individual users, user groups, or public. The interpretation is determined by the UserKind field of a rule. The types of rules, for illustration purposes, can be given the names ACL Rule for User, ACL Rule for Group, and ACL Rule for Public respectively.

By specifying Public, the ACL Rule for Public authorizes all the users to perform operations specified in the ACL Privileges on the bound entity, provided that the users pass their User Privileges check. The ACL privileges on the bound entity to Public can be configured in the System level. The capability of opening a bound entity to Public can be configured system-wide. The configuration parameter is named PubAccessEnabled (defined in table ICMSTSYSCONTROL). When disabled, all the ACL Rules for Public are ignored during the access control process.
Figure 8-9 shows the system administration dialog box through which public access can be enabled or disabled:

![Library Server Configuration](image)

**Note:** In Content Manager Version 8.3 *Public Access* is disabled by default. This means that all ACLs that specify Public Access are ignored during the access control list process.

If public access is enabled, both of the following ACL rules are checked:

- ACL rules of user / user group
- ACL rules of ICM public
In Content Manager Version 8.3, you set the ACL check level on the Access Control pane for the item type. You can select to check an ACL at the item type level or at the item level. For item level ACL checking, you can assign an ACL from the item type ACL, a user’s default ACL, or use an ACL provided by the application. See Figure 8-10.

![Figure 8-10 Define access control on item type](image)

Within the same ACL, a user can be specified in more than one type of rule. The precedence of the three types, from highest to lowest, is ACL Rule for Public, ACL Rule for User, and ACL Rule for Group. When applying ACL checks, if any higher-precedence rule type passes, the authorization is resolved and the process stops. If the check for ACL Rule for Public fails, the checking process continues on the lower-precedence rule types.
Tip: The Content Manager Client for Windows does not perform part-level ACL validation. Menu items such as Open, Browse, menu items related to annotations, and menu items related to the note log are not disabled even if the user does not have access to these parts. To avoid these problems, the Content Manager system administrator should associate an ACL (to each of the parts) that authorizes the user to read, add, and update the parts under the Document Management tab of the System Administration Client when creating the item type.

If the check for ACL Rule for the User failed, however, the checking stops. The ACL Rule for Group is not checked. There is no need to continue the check on the Group type because if a user does an individual user check, the user is excluded from the group type access based on the access control algorithm. The access control check for individual User type and Group type is not a sequential process, it is an either-or situation, even though there is no harm in doing a sequential check.

If the user fails to pass an individual user type check (or the user does not have a rule in the Access List table), the checking process continues to the group type. If the user belongs to one of the groups and the check of the privilege passes, the authorization is resolved and the process stops. Otherwise, access is denied and the process also stops. When a user is specified in more than one ACL Rule for a Group, the user is authorized by the union of all those rules’ ACL Privileges. A user is never specified in more than one ACL Rule for User.

Figure 8-11 shows the ACL algorithm for granting or denying access.
The Content Manager system provides the following pre-configured ACLs:

**NoAccessACL**  This ACL contains no rule, which is the same as saying that no one has any privileges.

**PublicReadACL**  This ACL consists of a single rule that specifies, for all Content Manager users (ICMPUBLC), the read capability (ClientUserReadOnly) is allowed. This is the default value assigned to a user’s DfltACLCode.

**SuperUserACL**  This ACL is reserved for future use.

**Assigning a privilege set to an access control list**

Each user ID that you add to an access control list (ACL) needs a privilege set associated with it. The user ID and privilege set define which users have access to an object and what kind of access they have to that object.

Users cannot access any object unless they are on the ACL. To add a user or user group to an ACL, you need to select a user ID and a privilege set for the ACL and click **Add**. For each defined ACL, the user IDs and groups are listed in the Access Control List window. You can modify this table by adding and removing user IDs and groups.
8.3.8 Access control list user exits

Content Manager provides two access control user exit routines that override the built-in access control mechanism.

For general privilege access control, Content Manager checks the privileges of the current user to determine whether the user has authority to perform the requested operation. For Item / View level access control, Content Manager dynamically checks for required privileges during the action being performed using SQL.

Each of these privilege checks has an exit routine to override Content Manager’s built-in access control mechanism. The exit routine names are:

ICMACLPrivExit  This exit routine can be used to determine whether a user has the authority to perform the requested function on a particular item or view. The member name for this exit is ICMACLXT.

ICMGenPrivExit  This exit routine can be used to determine whether the user has the general privilege to perform the requested function. The member name for this exit is ICMGENXT.

UDF declaration for ACL exits

For the complete UDF declaration for both the ICMACLPrivExit and the ICMGenPrivExit, see Appendix D, “ACL user exits UDF declarations” on page 665.

Enabling ACL user exits

Use the steps below in order to turn on ACL user exit checking:

1. Log on to the Content Manager System Administration Client, and select the Library Server for which you want to enable ACL user exit checking.
2. Open the Library Server Configuration dialog box.
3. Within the Features panel, check the Enable ACL User Exits check box (see Figure 8-12).

Important: Do not enable the ACL user exits until the exit DLLs are copied to the correct locations (see step 3 below).
Figure 8-12  Enabling ACL user exists on the Content Manager Library Server

**Compiling ACL exit routines**

**For AIX / SUN:**

1. Copy `<%ICMROOT%>/samples/server/exit/* to your working directory.
2. Use the sample exit program to implement your own authentication logic.
3. Modify the make file to reflect your environment:
   - set LIBDIR (<insthome>/sqlib/function)
   - set DB2_HOME (db2 install path)
   - set OBJDIR (object file directory)
   - set SRCDIR (working directory with source files)
4. Execute `make -f <makefilename>` from the working directory.

**For Windows:**

1. Copy `<%ICMROOT%>/samples\server\exit\* to your working directory.
2. Use the sample exit program to implement your own authentication logic.
3. Modify the make file to reflect your environment:
   - set LIBDIR (<db2path>\function)
   - set DB2_HOME (db2 install path)
   - set OBJDIR (object file directory)
   - set SRCDIR (working directory with source files)
4. Execute `nmake -f makefilename` from the working directory.
**Copying DLL**

After the two user exit routines are compiled, they need to be copied from the working directory to a location that Content Manager can use them. Here are the locations to copy the DLL's to:

- icmgenxt.dll → PATHICMDLL from the Library Server configuration dialog box
- icmacixt.dll → <db2instance_home>/sqllib/function

**Note:** These are the naming conventions:

- icmgenxt.c (general privilege exit program)
- icmacixt.c (ACL privilege exit program)
- icmn*mak (Windows platform make file)
- icmx*mak (AIX platform make file)
- icms*mak (SUN platform make file)

### 8.3.9 Domains

Administrative domain is a new feature in Content Manager V8. It is added so that the administrative tasks of managing users and access to stored objects can be delegated to more than a single system administrator. Domains allow a single Content Manager system to be used in support of multiple departments, or business areas, where an administrator from those areas has responsibility to manage the users accessing their domain. Essentially, all users in each area are using a single Content Manager Library Server and its associated Resource Managers, but with restricted access to the overall system.

A domain is a section of a Library Server that one or more administrators manage. Domains consist of user IDs, user groups, privilege sets, access control lists, Resource Managers, and SMS collections. Domains are not visible to users, so what you name your domains only have meaning to you and the system administrators who manage them. Users do not know that you have limited them to a part of the Library Server, meaning that they only know about items within that domain.

Domains limit administrative and user access to a subsection of the Library Server. An administrator with full privileges to the Library Server can delegate limited administrative privileges to another administrator. The administrator with full privileges, a super administrator, has access to all sections of a Library Server while an administrator with limited privileges, a sub-administrator, has access to only a section of the Library Server.

Sub-administrators can only view ACLs and privilege sets. Only super administrators can create, update, and delete ACLs and privilege sets.
A sub-administrator may share different combinations of the super administrator responsibilities but only for their domain. By creating domains and assigning administrators to manage those domains, the super administrators can delegate subtasks while concentrating on the overall system and managing it efficiently as the sub-administrators manage users and tasks specific to their domain.

Before you enable domains, consider the following conditions:

- Once enabled, administrative domains cannot be disabled.
- The three default domains cannot be modified.
- The administrative objects from the default domains cannot be deleted.
- The ICMPublic user group cannot be moved outside of the default public domain.
- Resource Managers, collections, user IDs, and user groups can exist in only one domain at a time.
- Privilege sets and access control lists can exist in more than one domain at a time.
- Except for the PUBLIC (shared) domain, domains do not overlap.
- Any object created in the super administrative domain cannot be moved, whether if it is system generated or user created.

To enable domains, go to the file menu, select Tools → Administrative Domains and then select Enable Administrative Domains (see Figure 8-13). You need to restart the System Administration Client for the domains to take effect.

**Important:** Once administrative domains is enabled, it cannot be disabled.

![Figure 8-13 Enabling Content Manager administrative domains](image)
Once administrative domains is enabled, the default domains listed in Table 8-4 are created.

Table 8-4 Default Content Manager administrative domains

<table>
<thead>
<tr>
<th>Default domains</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuperDomain</td>
<td>Used for super system administration access to all Content Manager services. The ICMADMIN user ID belongs to this domain by default and cannot be moved. In addition, all default privilege sets and Access Control Lists belong to this domain so an administrator with super system access can control all parts of the Content Manager system.</td>
</tr>
<tr>
<td>PublicDomain</td>
<td>Default used for public access to parts of the system. The ICMPublic user group belongs to this default domain and cannot be moved. In addition, this domain can be assigned to groups to insure that members in the group maintain the domain access defined in their individual user profiles.</td>
</tr>
<tr>
<td>DefaultDomain</td>
<td>Default domain designed to accommodate sub-administration or administrators responsible for a particular domain. When creating new users, this is the default domain most used unless specific domains are created.</td>
</tr>
</tbody>
</table>

Of course you can create your own domains, by using the Content Manager System Administration Client. When you create a domain and give it a name, you are merely creating a label. The definition of a domain is the privileges, access control lists, users, collections, and Resource Managers assigned to the domain.

Note: Once users, privilege sets, access controls, and other objects are assigned to a domain that you create, the domain cannot be deleted until all objects are re-assigned to another domain.

Administering domains

Depending on your privilege set, you administer either the entire Library Server or a specific domain. An administrator who has full access to the Library Server is a super administrator. A sub-administrator has full access to the objects in a specific domain.

Each type of administrator has the ability to create, retrieve, update, and delete the objects in their domains, including users and collections. Sub-administrators can see and retrieve objects only in their domain and list or retrieve in the PUBLIC, or shared, domain.
**Accessing domains**

Sub-administrators cannot change the domain of an object. They can, however, access the contents of their own domain and list or retrieve any object in the PUBLIC, or shared, domain.

Super administrators have access to all domains on the Library Server. They can create an object and assign it to a domain. Some objects, such as privilege sets and ACLs, only they can create for sub-administrators to use.

Sub-administrators can only do create, retrieve, update, and delete for any objects in their domain.

**Assigning a user to a domain**

When you create a user ID, you have the choice to assign it to a domain, or leave it in the default domain. You can change the domain of the user ID at a later time through user properties.

A user ID can have access to only one domain at a time. You cannot add a user to the PUBLIC, or shared, domain.

Only super administrators have the authority to create domains and assign users to those domains. A domain can have more than one sub-administrator, but only the super administrator can define who those administrators are by giving them system administration privileges within a privilege set. The Grant privilege set field in the New User or User Properties window indicates which administrative privileges a sub-administrator has within a domain.

**Assigning a user group to a domain**

Assigning a user group to a domain changes the domain designated for each user ID in that user group. A user ID can have access to only one domain at a time. So, any user ID included in a group that you assign is also moved to the new domain. A user group name can be in only one domain at a time. You can assign the user group into the PUBLIC, or shared, domain, but not to the super domain.

**Assigning a privilege set to a domain**

Any user ID that you add to a domain must also have an associated privilege set. If you do not include the associated privilege sets, then the users cannot perform their tasks. The best place to store privilege sets to make them available to any user is the PUBLIC, or shared, domain.
Assigning a Resource Manager to a domain

You can restrict user access to certain Resource Managers by assigning them to a specific domain. When you define a new Resource Manager for a Library Server to access, you have the option to select a domain.

The default for all Resource Managers is PUBLIC. If you do not want everyone to have access to the Resource Manager, you need to assign it to a domain. If you do not see a domain that you can assign the Resource Manager to, you can still define the Resource Manager and then create the domain you need. After you have the appropriate domain defined, open the Resource Manager properties and select the domain.

Assigning a collection to a domain

You can restrict user access to a certain collection on a Resource Manager by assigning it to a specific domain. If the Resource Manager is in the PUBLIC domain, you can assign a collection to any other defined domain. If the Resource Manager, however, is defined to a specific domain already, then you cannot assign the collection to another domain, even if you want to assign the collection to the PUBLIC domain.

A user needs access to the Resource Manager to access the collections on it, so you cannot restrict access to the Resource Manager without imposing the same restrictions to the collections on it.

8.4 Access to objects

The Resource Manager is the repository for content stored in the Content Manager system. Users store and retrieve objects to and from the Resource Manager by issuing requests through the Library Server. When a request is granted, the Library Server returns a security token and the location of the object to the user.

In Content Manager V8, the Library Server is the entity which knows the rules which govern access to objects. As the objects are actually stored on Resource Managers, security tokens provide a mechanism through which a Resource Manager can determine if the Library Server has permitted access to a particular object. When retrieving content, the client uses this security token to access the Resource Manager, and to give the location of the object. The object is then returned to the client.
The Client now becomes the central cohesive element in fetching objects from the Resource Manager. In general, there is very little communication between the Library Server and the Resource Manager. The client communicates with the Library Server through the database query language SQL, to find and request access to data. When the Library Server authorizes the access, it sends a security token related to the asset directly back to the Client unlike previous Content Manager implementations. The Client then directly communicates with the Resource Manager using the Internet protocols HTTP, FTP, and FILE. This new design configuration leads to the so called inverted “V” design. The Client sits at the apex of the inverted “V” and the Library Server and Resource Manager sit at the base points.

**Security tokens**

In the V8 Content Manager architecture, a client first retrieves an object (parts) ID from a Library Server and then uses the object ID, through the Resource Manager API, to retrieve the object from a Resource Manager. This is commonly referred to as the pull model. In general, the object ID alone does not provide adequate access control protection. With the pull model, access control is performed jointly between the Library Server and Resource Manager.

When a Library Server receives a client request for accessing an object stored in a Resource Manager, it first determines whether the client is authorized for the requested action. If it is, the granted access is returned to the client in the form of a security token along with the requested object ID. The client then accesses the object by presenting the token to the Resource Manager. In turn, the Resource Manager checks the client’s access privilege by first validating the access token before honouring the request.

**Security token generation**

Tokens are generated by Library Servers and are validated by Resource Managers. Library Servers and Resource Managers share a secret symmetric encryption key. The tokens contain a message authentication code (MAC), which provide a fingerprint for a particular access to a given object. The MAC is a one-way hash function with the addition of an encryption key.

The function `GenToken()` is invoked by the Library Server to generate a security token for object access control in the new Content Manager pull model. `GenToken()` takes the hostname or IP address, object ID (itemID and version number), token expiration time, requested access type (create, read, update, delete, stream) and encryption key as input and returns an access token as output.
The security token is generated by computing the MAC for the given combination of Resource Manager/objectID/access type/expiration time/encryption key via an algorithm. Also included explicitly in the token are the access type permitted and the expiration time, as the Resource Manager must know this information in order to validate the token. The security token is always encoded into a printable character string, as shown in bold within example Example 8-2, which is a URL that a client is passing to a Resource Manager in order to access an object:

Example 8-2  URL passed from a client to a Resource Manager for object access


Security token validation
The Validate_token() function is invoked by a Resource Manager to check whether an object access request by an application should be granted or not. If the token passed to the Resource Manager with the object name is valid, the request is granted; otherwise, it is rejected. When an application issues ICMRetrieveObj() to retrieve an object, for example, it passes in 'object name, access token, access type' as parameters among other things. ICMRetrieveObj() function first validates the application's access permission by calling Validate_token(). If Validate_token() returns OK, access is granted; otherwise, the ICMRetrieveObj() request is rejected.

The Validate_token() function checks whether the token is valid, which includes checking the expiration time of the token and the access type. If the token is valid and has not expired, it returns an OK return code to the calling function, in this case ICMRetrieveObj(). If the token is expired, then an “expired” return code is returned. If the access type requested is not allowed, then an “invalid access” return code is returned. If the token is not valid for any other reason, then a “retry” return code is returned. In the case of a “retry,” the application can make a call to the Library Server to obtain a new security token in case the encryption key is flushed.
**Encryption key management**

The token generation and verification functions rely on an encryption key. The encryption key is Library Server specific and is a shared secret between a Library Server and all Resource Managers. The encryption key of a Library Server can be refreshed in case the key has been compromised; see Figure 8-14.

![Figure 8-14 Refreshing Library Server encryption key](image)

**Important:** For security reasons, we recommend that you periodically refresh the encryption key (especially in situations where your clients include Internet users). When it is refreshed, a new key is used to decrypt the security tokens.

When encryption keys are regenerated, the Library Server first generates a new set of keys. The Library Server then sends this information to the Resource Managers. If a Resource Manager is not running or functioning correctly, then it cannot obtain the new set of keys. The regeneration of the encryption key should only take place when all Resource Managers are up and running.
If a Resource Manager is not able to obtain the new keys, it throws an error message similar to the one below when a client requests a document (see Figure 8-15).

![Error message](image)

\textit{Figure 8-15 Error message when Resource Manager encryption key is out of date}

In this situation, you simply need to start the Resource Manager that you were having trouble retrieving objects from, and then refresh the encryption key again from the Library Server Configuration window (making sure any other Resource Managers you have are also available!).

It should also be noted that once a Resource Manager obtains a new encryption key, it cannot decrypt tokens which may have been previously given to clients. As a result, even if the old tokens are not yet expired, clients need to obtain a new token from the Library Server, which uses the new encryption key.

\textbf{Token duration}

Token duration specifies how long the security token, provided to a client to access a document on the Resource Manager, remains valid. The default is 48 hours (172800 seconds). A client application may continue to access the resource object without reference to the Library Server for the period that the token remains valid. If your application frequently re-reads the same item on the Resource Manager, you can store the item’s URL and go directly to the Resource Manager for the item for as long as the token remains valid.

The token duration value can be updated for each Resource Manager through the Content Manager System Administration Client. Go to the Resource Manager properties window, and enter the time \textit{in seconds} that you wish the tokens generated for a particular Resource Manager to remain valid (see Figure 8-16).
Figure 8-16 Setting the Resource Manager token duration value

Once a token expires, it can no longer be reused and the client must request a new token from the Library Server. A token should always be allowed to expire. This is because once a client has obtained a token to retrieve an object, the token can be used to access the object even after the access control in the Library Server has been modified to restrict access. Unless there is a particular need for a longer token duration period, (such as in a custom web application which dynamically generates links for documents) the default value is acceptable in most situations.

8.5 WebSphere global security

The Resource Manager is a Web application, and is deployed onto WebSphere Application Server. When WebSphere Application Server is first installed, there is no security mechanism enabled. It is possible for an individual to access your WebSphere Administrative Console using the address:

http://<hostname>:9090/admin
Any individual can log on into the console without a password (see Figure 8-17), and have full control over your WebSphere environment. If you are installing a new WebSphere environment for Content Manager, it is important to enable global security. This should be done before installing Content Manager (although it can be enabled afterwards) so that your Resource Manager applications are secure right from the start.

**Note:** If you have an existing WebSphere Application Server environment, it is extremely likely that global security has already been enabled.

![Figure 8-17 Logging on to the Administrative Console with global security disabled](image)

After global security is enabled, the communications to your WebSphere Application Server Administrative Console uses https, and you are required to enter a user ID and password in order to access the console (see Figure 8-18). The user ID and password you enter can be defined within the local operating system, an external LDAP directory, or a custom user registry that implements the com.ibm.websphere.security.UserRegistry interface, depending on which active user registry is selected and configured while enabling WebSphere global security.
**Note:** When you enable security, you are enabling security settings on a global level. When security is disabled, WebSphere Application Server performance is increased between 10-20%. Consider disabling security when it is not needed. For example, when you operate behind a very secure firewall, or in a testing environment, where security is not so crucial, you may want to consider disabling the security.

Notice in Figure 8-18 that https is now used as the communication protocol when connecting to the WebSphere Application Server Administrative Console.

Global security applies to all applications running in the environment and determines whether security is used at all, the type of registry against which authentication takes place, and other values, many of which act as defaults.

The term *global security* represents the security configuration that is effective for the entire security domain. A *security domain* consists of all servers configured with the same user registry *realm* name. In some cases, the realm can be the machine name of a Local OS user registry. In this case, all application servers must reside on the same physical machine. In other cases, the realm can be the machine name of an Lightweight Directory Access Protocol (LDAP) user registry.

Figure 8-18  **Logging on to the Administrative Console with global security enabled**
Since LDAP is a distributed user registry, a multiple node configuration is supported, such as the case for a Network Deployment environment (Content Manager comes with “base” WebSphere Application Server, not the Network Deployment version). The basic requirement for a security domain is that the access ID returned by the registry from one server within the security domain is the same access ID as that returned from the registry on any other server within the same security domain. The access ID is the unique identification of a user and is used during authorization to determine if access is permitted to the resource.

Configuration of global security for a security domain consists of configuring the common user registry, the authentication mechanism, and other security information, which defines the behavior of a security domain. The other security information that you can configure includes Java 2 Security Manager, Java Authentication and Authorization Service (JAAS), Java 2 Connector authentication data entries, Common Secure Inter-operability Version 2 (CSIv2)/Security Authentication Service (SAS) authentication protocol (Remote Method Invocation over the Internet Inter-ORB Protocol (RMI/IIOP) security), and other miscellaneous attributes. The global security configuration usually applies to every server within the security domain.

WebSphere global security is enabled via the WebSphere Application Server Administrative Console (see Figure 8-19). After enabling global security, you need to enter a user ID and password in order to perform some functions from the command line, such as starting and stopping application servers, for example:

```
stopserver server1 -user admin -password password
```
Chapter 8. Security

Figure 8-19 Enabling WebSphere global security via the Administrative Console

**Important:** When WebSphere global security is enabled, it is very important to correctly fill in the installation panel for automatically deploying the Resource Manager application to WebSphere Application Server V5 during the installation of a Resource Manager. This is because the user ID and password that you enter on this panel is used by the installation process to access WebSphere Application Server in order to deploy the Resource Manager Web application. If either value is incorrect, the deployment fails.

For full instructions on how to enable WebSphere global security, and for further background on the options available, refer to the IBM provided manual *IBM Websphere Application Server v5 - Security.*
8.5.1 Java 2 security

Java 2 security provides a policy-based, fine-grain access control mechanism that increases overall system integrity by checking for permissions before allowing access to certain protected system resources. Java 2 security guards access to system resources such as file I/O, sockets, and properties. J2EE security guards access to Web resources such as servlets, JavaServer pages (JSPs) and EJB™ methods. WebSphere global security includes J2EE role-based authorization, the Common Secure Inter-operability Version 2 (CSIv2) authentication protocol, and Secure Sockets Layer (SSL) configuration.

Java 2 security is disabled by default in WebSphere Application Server V5, however it is enabled automatically when you enable WebSphere global security when configuring security. Although it becomes enabled automatically when you enable WebSphere global security, you can choose to disable it. You can configure Java 2 security and global security independently of one another. Disabling global security does not disable Java 2 security automatically. You need to explicitly disable it.

The Resource Manager application is Java 2 security ready; therefore, you can leave Java 2 security enabled (see Figure 8-20) when you enable WebSphere global security, prior to installing Content Manager V8.2.

![Figure 8-20 Enforcing Java 2 security using the Administrative Console](image-url)
8.6 Content Manager and RACF

Almost all of the information provided in the previous sections of this chapter also applies to z/OS. There are, however, some differences that you must take care of during the installation and use of Content Manager for z/OS. The following information provides an understanding of how things work on z/OS and RACF.

8.6.1 User IDs

During the installation of Content Manager for z/OS, several different user IDs are required. If you have set up your system to use server security, the user ID and password is checked against the information stored in your host security system.

*Installation user*

During the installation of Content Manager for z/OS Version 8.2, there is only one user ID necessary. This user ID must have proper authority to perform the required DB2 tasks and to update the definitions for the HTTP server. To do this, the user ID must have an OMVS segment defined.

These are the DB2 actions performed during the installation:

- Create STOGROUP (or, if using an existing storage group, the user needs the USE OF STOGROUP privilege)
- Create:
  - Database
  - Tablespace
  - Table
  - Index
  - View
  - Procedure
  - Function
- Bind:
  - Package
  - Plan
- Grant execute of:
  - Package
  - Plan
  - Procedure
  - Function
- Insert into table
- Alter tablespace
These are all DB2 actions performed during the installation of Library Server and Resource Manager. If you have a problem during the installation process, it may be necessary to drop your database and the procedures and functions. As the creator of these objects, this should be no problem. If the database definition was created by a DB2 system administrator other than you, this person has to perform the drop and the new creation.

**Usage**

To work with the Content Manager for z/OS Version 8.2, you need to have three different user IDs:

- Content Manager administration user
- DB2 Connect™ user
- Resource Manager administration user

At least these three users must be defined in RACF. Each of them is dedicated to a special task. See the previous discussion in this chapter to get more information about this.

We recommend that you use three different RACF user IDs, each set up in Content Manager with the appropriate set of privileges.

**Content Manager administration user**

This is the administration user for the Content Manager system. This ID is created in the Content Manager tables during the installation. The default Content Manager password for this user ID is "password". If you have enabled server security, the password is checked against the password for this user ID in your z/OS security system, for example RACF.

If you have enabled the security exit program, the password is checked against whatever the exit program does. See 8.6.2, "RACF user exit" on page 232 for more information about the RACF user exit.

**DB2 Connect user**

The employees of business departments normally are not defined as DB2 users. They work on the company databases using the Content Manager application. Because the Library Server is a set of DB2 stored procedures and functions, it is necessary that each user is connected to the DB2 subsystem before the user can perform any function. Even for the logon to Content Manager, you must connect to the DB2 of the Library Server. To do this, a file with the DB2 Connect user ID and password (both encoded) is stored on every Windows client. The name of the file is cmbicmenv.ini and it is stored in the CMgmt directory. If the user ID which wants to log on does not have the DB2 privilege to connect to the database, then the values of this files are taken by the client. This file is created during the installation of the Windows client.
For security reasons, the DB2 Connect user ID should be limited to the rights to logon to the DB2 system. This mechanism is the same on every platform.

You can define several different DB2 Connect user IDs.

**Resource Manager administration user**

In the 390 environment, nobody really “logs on” to the Resource Manager. All requests send to the Resource Manager are HTTP requests send to the HTTP server running on the z/OS system. These requests are built from the Library Server, sent to the workstation, and from the workstation, sent to the Resource Manager.

In the HTTP request sent from the Library Server to the workstation, there is the Resource Manager administration user ID and password (encoded) included. The Library Server takes these values from the ICMSTResourceMgr table.

The z/OS Resource Manager checks this authorization against RACF. It must be a valid RACF user ID. If not, an HTTP authorization failed message is returned to the client.

See Figure 8-21 for a principal flow overview.
8.6.2 RACF user exit

The RACF user exit is called during logon to the Content Manager. The user ID and password are passed to the exit and can be checked against RACF or any other security system installed on your system. Also, the password can be changed in this exit program.

Sample code is provided in dataset SICMSAM1. The name of the exit program is ICMXLSLG. This is a C sample. In the sample code, another program, named ICMMRACF, is called. This is an Assembler program which is actually the interface to RACF.

**Note:** There is no check box to activate the RACF user exit with the administration client. During logon, the stored procedure tries to find the ICMXLSLG program. If it is not there, normal processing continues. If it is found, the exit is called and all other actions must be taken by the exit. To activate the exit, put the load module in any of the steplib libraries of the WLM of your Library Server or add the library to the steplib concatenation.
8.6.3 RACF import utility

The RACF import utility provides an easy way to import users who are already RACF defined into your Content manager database. It consists of a set of jobs that can be found in SISMINS1.

Figure 8-22 gives you an overview of the job flow.

Figure 8-22   RACF batch import utility

Here is a more detailed description:

- **ICMMBKUP**
  This JCL backs up RACF information into the RACFBKUP file.

- **ICMMMDATA**
  This JCL reads the RACFBKUP file to dump RACF user information into the RACFDATA file.
- **ICMMSORT**
  This JCL selects three fields: record type 0220 (user TSO data), users, and groups from RACFDATA. It sorts the data, and stores sorted information in RACFSORT.OUT.

  **Note:** Here you have the chance not only to sort but also to extract which user you want to be selected.

- **ICMMUSTB**
  This JCL queries Content Manager table (icmstusers) to get all Content Manager users and stores in USERTABL.OUT.

- **ICMMCOMP**
  This JCL requires RACFSORT.OUT and USERTABL.OUT as input files and generates RESCOMP.OUT. (RESCOMP.OUT has list of users with either 'A' or 'D' appended to indicate new Content Manager users to be added, or existing Content Manager users to be deleted, respectively).

  The load module, icmxcomp, is required to execute this job successfully.

- **ICMMDFUR**
  This JCL requires RESCOMP.OUT as the input file. The default user information is included in this JCL. At this point, please review RESCOMP.OUT to confirm the users are added and deleted.

  This is the final JCL that defines and deletes Content Manager users.

All of the above JCLs provide the flexibility to be configured in multiple ways per business requirement. This is one such sample requirement.

**Note:** If using the import utility, you also have to activate the RACF exit program, because no passwords are transferred. Otherwise, you have to enter all the passwords through the administration client.
Tivoli Storage Manager for Content Manager

In this chapter, we provide an overview of IBM Tivoli Storage Manager (TSM), including a description of its architecture and components. There is also a section which gives advice on when IBM Tivoli Storage Manager should be used with Content Manager, and a high level description outlining the procedure to integrate the two products.
9.1 IBM Tivoli Storage Manager

IBM Tivoli Storage Manager (TSM) is an enterprise-class storage and recovery solution, protecting business critical data from a wide diversity of systems. TSM is a client/server program that provides an automated, centrally scheduled, policy managed backup, archival, and space management facility for file servers and workstations.

TSM ensures availability of business applications by providing data protection and resource utilization that scales with business needs. To protect these business needs, TSM integrates the power of application-aware technology in the recovery of leading database, content management, and workflow applications to ensure the entire business-process is protected.

In a Content Manager environment, TSM can be used to manage the migration of data through its lifecycle, from creation to deletion or disposition. Through the integration with TSM, the Content Manager administrator can use a wide variety of storage technologies and create a hierarchy of devices. Content is migrated between the devices with no intervention from the administrator, and no impact on the user. This allows newer, more frequently used data to be placed on faster, typically more expensive media; and older, archival data to be placed on slower, cheaper media.

9.1.1 Overview of Tivoli Storage Manager capabilities

Tivoli Storage Manager supports a wide variety of storage devices, some of which include SAN attached hard disk subsystems, manual and automated tape devices, and optical jukeboxes. This provides Content Manager systems the ability to support storage devices other than fixed disks attached to the Resource Managers.

TSM provides the following capabilities, among others:

- Backup and restore
- Archive and retrieve
- Disaster preparation and recovery

Note: For information on Content Manager backup, recovery, and high availability, refer to the following IBM Redbook:

- Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures, SG24-7063
Backup and restore
Backups are copies of the active online data stored on offline storage. The backup process copies data from client workstations to server storage to ensure against loss of data that is regularly changed. The server retains versions of a file according to policy, and can replace older versions of a file with newer versions. Policy includes the number of versions to keep and the retention time for the versions.

If an online storage device should fail, a data error occurs, or someone accidentally deletes a file, the offline copy of that data can be copied back to online storage. A client can restore the most recent version of a file, or can restore to an earlier version.

Backups can be scheduled, performed manually from the TSM client interface, or performed remotely using a Web-based interface.

TSM provides backup and restore functionality, using multiple techniques to reduce data transfer sizes to the minimum possible. These techniques reduce the total time required for both data backups and more importantly, data restores.

Archive and retrieve
The archive process copies data from client workstations to server storage for long-term storage. The process creates a copy of a file or set of files and stores it as a unique object for a specified period of time. Archiving inactive data can be an effective way of reducing storage costs.

Files can remain on the local storage media, or can be deleted. The server retains archive copies according to the policy for archive retention time.

TSM also provides retrieval functionality so that the previously archived data can be retrieved. The retrieval process locates the copies within the archive storage and places them back into their original location by default, or to a new destination if specified.

Disaster preparation and recovery
A very important part of data protection is disaster preparation and recovery. Local copies of data protect against discrete failures or errors in equipment, storage, or people; however, disasters tend to happen to the entire facility, not just a section of the equipment inside the facility.

Using TSM, you can prepare an additional copy of the active data for safekeeping at an off-site location, to provide extra insurance against disasters. Should a disaster strike and destroy the online storage and computers, the off-site copy of the active data can be restored to new computers to get business up and running quickly.
9.1.2 Architecture

Tivoli Storage Manager employs a unique architecture: network storage management with relational database and recovery log, specifically designed for managing the data storage needs of complex computing environments. The TSM architecture provides a single, sophisticated storage management system that can be exploited by virtually all storage management applications. It supports not only backup and recovery applications, but also disaster recovery, hierarchical storage management, archiving, document management and the management of data objects generated by custom or internally developed applications.

TSM is implemented as a client/server software application, which consists of a server software component, backup/archive client, and other complementary Tivoli and vendor software products.

The TSM server provides a secure environment, including automation, reporting and monitoring functions, for the storage of client data. It also provides the storage management policies and maintains all object inventory information.

The TSM client software and complementary products implement data management functions such as data backup and recovery, archival, hierarchical storage management, or disaster recovery.

The client software can run on different systems, including laptop computers, PCs, workstations, or server systems. The client and server software can also be installed on the same system for a local backup solution.

**Tivoli Storage Manager server**

One of the principal architectural components of the TSM server is its built-in relational database. The TSM database was specially designed for the task of managing a data storage environment, and it implements zero-touch administration. The server database operates transparently, requiring minimal administrative oversight. This database is fully protected with software mirroring, roll-forward capability, and with its own management and online backup and restore functions.

The TSM server uses this database to intelligently map business goals with storage management policies and procedures. The TSM server tracks the origin and location of each client data copy. Policies defined to the TSM server determine how data copies are stored, migrated, and eventually replaced with newer data.

All database transactions are written to an external log file called the recovery log. The TSM server uses the recovery log as a scratch pad for the database, recording information about client and server actions while the actions are being performed. The recovery log can be used to restore the database if necessary.
For storing the managed data, the TSM server uses a storage repository. The storage repository can be implemented using any combination of supported media: magnetic or optical disk, tape, and robotic storage devices, which are locally connected to the server system or which are accessible through a SAN. To exploit SAN technology, the TSM server has features implemented to dynamically share SAN connected automated tape library systems among multiple TSM servers.

The TSM server provides built-in device drivers for more than 300 different device types from every major manufacturer. It is also able to utilize operating system device drivers and external library manager software.

Within the storage repository, the devices can operate stand-alone or can be linked together to form one or more storage hierarchies. The storage hierarchy is not limited in the number of levels and can also span over multiple servers using virtual volumes.

**Backup and archive client**

Data is sent to the TSM server using the TSM backup/archive client. These clients work together with the TSM server base product to ensure that the data you need to store, is managed as defined. The TSM backup and archive client, included with the server, provides the operational backup and archival function. The client implements the patented progressive backup methodology, adaptive sub-file backup technology, and unique record retention methods. The TSM client must be installed on every machine that transfers data to server-managed storage.

The TSM server uses a unique node name to identify each TSM client instance. A password can be used to authenticate communications between the TSM client and server. Data can be recovered to the same client machine that initially transferred it, or to another client with a compatible file system format.

The TSM client basically consists of the software component and a customization file. This customization file, called the client options file, specifies client/server communications parameters and other TSM client settings. Client communications parameters must agree with those specified in the server options file. The client options file is located in the client directory and can be modified using a text editor. The client graphical interface also provides a wizard for editing this file.

Some client options can also be defined in the TSM server database. Defining these client option sets allows for the centralized management of certain client operations.
The backup and archive clients are implemented as multi-session clients, which means that they are able to exploit the multi-threading capabilities of modern operating systems. This enables the running of backup and archive operations in parallel to maximize the throughput to the server system.

Depending on the client platform, the backup and archive client may provide a graphical, command line or Web user interface. Many platforms provide all three interfaces. The command line interface is useful for experienced users and allows the generation of backup or restore scripts for scheduled execution. The graphical interface is designed for ease of use for the end user, and ad hoc backups and restores. The Web client is especially useful for those clients, such as NetWare, where no native GUI is available, or for performing remote backup and restore operations, for example, in a help desk environment.

Some clients (including some UNIX variants and Microsoft platforms) use a new plug-in architecture to implement an image backup feature for raw device backup. This allows you to back up and recover data stored in raw (that is, not a file system) volumes. It also provides an additional method to make point-in-time backups of entire file systems as single objects (image backups) and recover them in conjunction with data backed up by using the progressive backup methodology.

**Administration**

TSM server operations are configured, controlled, and monitored using graphical or command line interfaces. Some tasks can be performed in several different ways, so the interface you use depends on the type of task and your preferences.

All policy information, logging, authentication and security, media management and object inventory is managed through the TSM server database. Most of the fields are externalized through the TSM high level administration commands, SQL SELECT statements, or for reporting purposes, by using an ODBC driver.
Many important TSM server functions can be automated through the use of schedules. A comprehensive and integrated set of schedules can provide the basis for efficient data management with little need for intervention during normal operations.

To schedule TSM server operations, you only need to create a schedule or set of schedules on the TSM server. Any of the following tasks can be automated:

- Backup and restore
- Archive and retrieve
- Tivoli Storage Manager server administration commands
- Running administrative scripts and macros

For the central administration of one or more TSM server instances, TSM provides command line or Java-based administration interfaces, also called administration clients.

Using the unique enterprise administration feature, it is possible to configure, monitor and manage all server and client instances from one administrative interface, known as the enterprise console. It includes:

- Enterprise configuration
- Administrative command routing
- Central event logging functions

The enterprise configuration allows TSM server configurations to be defined centrally by an administrator and then propagated to other servers.

Administrative command routing allows administrators to issue commands from one TSM server and route them to other target servers. The commands are executed on the target servers, and the command output is returned and formatted on the server where the command was issued.

In an enterprise environment with multiple TSM servers, client and server events can be logged to a central management server through server-to-server communications, thereby enabling centralized event management and automation.
**Externalized interfaces**

TSM provides a data management API, which can be used to implement application clients to integrate popular business applications, such as databases or groupware applications. The API also adheres to an open standard (XBSA) and is published to allow customers, or vendors, to implement specialized or custom clients for particular data management needs, or non-standard computing environments.

The API includes function calls that you can use in an application to perform the following operations:

- Start or end a session
- Assign management classes to objects before they are stored on a server
- Back up or archive objects to a server
- Restore or retrieve objects from a server
- Query the server for information about stored objects
- Manage file spaces

**9.1.3 Policy objects**

A TSM environment consists of three basic types of resources:

- Client systems
- Data
- Rules

The client systems generate the data, and the rules specify how that data are managed. For example, in the case of a TSM archive operation, rules define how long archived files should be kept and where they should be stored. Figure 9-1 gives an overview of the TSM policy objects.
Figure 9-1  TSM policy objects in action
TSM uses policy to define the relationships between these three resource categories. Depending on your needs, TSM policy can be fairly simple or more complex.

TSM policy objects can be divided into two interrelated groups:

- The policy objects that map to your business environment and data management goals
- The policy objects that map to your storage media and devices

To understand TSM data management policy objects, consider how they can reflect the organizational structure of your business environment. Table 9-1 introduces the TSM data management policy hierarchy, and provides examples of how you can use these policy objects to achieve your administrative goals.

**Table 9-1  TSM policy objects in the real world**

<table>
<thead>
<tr>
<th>TSM policy objects</th>
<th>Description</th>
</tr>
</thead>
</table>
| Policy domain      | You can map to different categories of TSM client nodes within your organization.  
For example, you may set up different policy domains for UNIX-based file server machines and Windows-based workstations. These domains can be used to provide customized storage management and separate administrative control for each logical group. |
| Policy set         | You can use policy sets to create subsets of TSM client nodes within a domain, but only one policy set can be active within a given policy domain at any time. Due to this restriction, many administrators implement just one policy set and focus their management effort on policy domains, management classes, and copy groups. |
| Management class   | You can map to different categories of data generated by your TSM client nodes. A management class contains one backup copy group, one archive copy group, or one of each. One management class in a policy set must be designated as the default. Additional management classes can be created and specified for use by individual TSM clients.  
For example, within the active policy set for the domain created for UNIX server machines, you may set up one management class for general data (default) and one for directory structure information. |
### Table 9-2   TSM media and device policy objects

<table>
<thead>
<tr>
<th>TSM policy objects</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy group</td>
<td>The working elements of TSM policy are defined in copy groups. These elements include the number of versions of TSM client files to be maintained and the amount of time those files are stored. The other TSM data management policy objects are primarily used to provide implementation flexibility. There are two kinds of copy groups: backup and archive. For example, within the default management class created to handle general data for the UNIX server policy domain, you may set up a backup copy group that maintains three copies of existing data and stores those copies for 100 days. By default, backup data for any TSM client nodes associated with this domain is managed to these specifications.</td>
</tr>
<tr>
<td>Volume</td>
<td>Represents one physical or logical unit of storage media. For example, a volume can represent a tape or a disk partition. Each volume is associated with a single storage pool. TSM classifies its volumes into two categories: private and scratch.</td>
</tr>
<tr>
<td>Storage pool</td>
<td>Represents a collection of available storage volumes of the same media type. TSM stores all managed data objects in storage pools. Storage pools are typically arranged in a hierarchy, with data migrating from one type of storage to another. For example, a storage pool with an LTO tape device class consists of a number of LTO tape volumes. Clients that need to back up data directly to LTO tape are associated with this storage pool. Other client data may go first to a DISK storage pool, and then migrate to the LTO storage pool.</td>
</tr>
</tbody>
</table>

#### 9.1.4 Storage devices and media

To store and manage data objects on various kinds of storage media and devices, TSM implements several logical entities to classify the available storage resources. Table 9-2 describes the TSM media and device policy objects.
<table>
<thead>
<tr>
<th>TSM policy object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device class</td>
<td>Represents the type of storage device that can use the volumes defined to a given storage pool. For example, an LTO tape device class can be used to associate a storage pool with any library device that handles LTO tape.</td>
</tr>
<tr>
<td></td>
<td>A device class specifies a device type and media management information, such as recording format, estimated capacity and labeling prefixes.</td>
</tr>
<tr>
<td></td>
<td>A device type identifies a device as a member of a group of devices that share similar media characteristics. For example, the 8MM device type applies to 8mm tape drives. Device types include a variety of removable media types, such as REMOVABLEFILE for Jaz or Zip drives, and also FILE and SERVER.</td>
</tr>
<tr>
<td></td>
<td>Magnetic disk devices are the only random access devices. All disk devices share the same device type and predefined device class: DISK. All other device types are sequential access devices.</td>
</tr>
<tr>
<td></td>
<td>Each removable media-type device class is associated with a single library.</td>
</tr>
<tr>
<td>Library</td>
<td>Represents a specific storage device.</td>
</tr>
<tr>
<td></td>
<td>For example, a library can represent a standalone drive, a set of standalone drives, a multiple-drive automated device, or a set of drives controlled by an external media manager.</td>
</tr>
<tr>
<td>Drive</td>
<td>Each drive mechanism within a device that uses removable media is represented by a drive object. Each drive is associated with a single library.</td>
</tr>
<tr>
<td></td>
<td>TSM drives include tape and optical drives that can stand alone or can be part of an automated library. Supported removable media drives also include removable file devices such as re-writable CDs.</td>
</tr>
</tbody>
</table>
9.1.5 Storage hierarchy and data migration

The storage pool is the central element of the TSM storage management environment because it provides the link between TSM data and storage objects. TSM allows you to organize storage pools into one or more hierarchical structures. Each storage hierarchy can span multiple TSM server instances.

Storage policy is used to migrate data objects automatically from one storage pool to another. This process helps to ensure that there is sufficient free space in the storage pool at the top of the hierarchy, where faster devices can provide the most benefit to clients. For example, the server allows you to initially back up data to fast storage media like disk, and then migrate the data to slower, less expensive media like tape during off-peak hours. It is possible to control when data migration begins and ends and also how the server chooses files to migrate.

**How the server groups files before storing**

When a user backs up or archives files from a client node, the server may group multiple client files into an aggregate of files. The size of the aggregate depends on the sizes of the client files being stored, and the number of bytes and files allowed for a single transaction.

**Where the files are stored**

When a user backs up, archives, or migrates a file from a client node, the server looks at the management class that is bound to the file. The management class specifies the destination, and the storage pool in which to store the file. The server then checks that storage pool to determine the following conditions:

- If it is possible to write file data to the storage pool (access mode).
- If the size of the physical file exceeds the maximum file size allowed in the storage pool. For backup and archive operations, the physical file may be an aggregate or a single client file.
- Whether sufficient space exists on the available volumes in the storage pool.

---

<table>
<thead>
<tr>
<th>TSM policy object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path</td>
<td>Represents a data and control path from a source to a destination. Paths allow access to drives and libraries. A path definition specifies a source and a destination. The source accesses the destination, but data can flow in either direction between the source and destination. To use a library or drive with TSM, a path must be defined between the device and either the TSM server or another designated data mover.</td>
</tr>
</tbody>
</table>
What the next storage pool is, if any of the previous conditions prevent the file from being stored in the storage pool that is being checked.

Using these factors, the server determines if the file can be written to that storage pool or the next storage pool in the hierarchy.

**Using copy storage pools to back up storage hierarchy**

For an additional level of data protection, copy storage pools can be used to back up your primary storage pools.

It is recommended, for efficiency, that you use one copy storage pool to back up all primary storage pools that are linked to form a storage hierarchy. By backing up all primary storage pools to one copy storage pool, you do not need to recopy a file when the file migrates from its original primary storage pool to another primary storage pool in the storage hierarchy.

A single copy storage pool can be used for the backup of all primary storage pools in most cases; however, in certain circumstances, multiple copy storage pools may be needed. For example, if you want to create multiple copies of your data to be written to multiple locations (to keep one copy on-site and one copy off-site for instance), then you need more than one copy storage pool.

### 9.1.6 Media management

TSM allows you to control how removable media are used and reused. After TSM selects an available medium, that medium is used and eventually reclaimed according to its associated policy.

TSM manages the data on the media, but you manage the media itself, or you can use a removable media manager. Regardless of the method used, managing media involves creating a policy to expire data after a certain period of time or under certain conditions, moving valid data onto new media, and the reuse of empty media.

**Tape rotation**

By providing policy objects that focus your management effort on data instead of media, TSM can help you fill in the gaps inherent in any tape rotation scheme. Instead of setting up a traditional tape rotation, you set up policy. Tape rotation, as it applies to TSM, can be thought of as the ongoing automated circulation of media through the storage management process. Once TSM selects an available tape, the tape is used and eventually reclaimed according to its associated policy.
Policy-based storage management takes a little time to understand and implement, but it allows for a great deal of automation and flexibility. Automating backup and recovery functions reduces the likelihood of human error, and also helps enforce data management goals.

In this section, we provide a general overview of the capabilities of TSM and its components and architecture. Most of the material provided is an extraction from existing TSM publications. If more detailed information is required, please refer to the following documentation and the Web site:

- IBM Tivoli Storage Manager Version 5.1 Technical Guide, SG24-6554
- IBM Tivoli Storage Management Concepts, SG24-4877
- IBM Tivoli Storage Manager: A Technical Introduction, REDP0044
- IBM Tivoli Storage Manager for AIX - Administrator's Guide, GC32-0768
- IBM Tivoli Storage Manager for AIX - Quick Start, GC32-0770
- The Tivoli Storage Manager home page, for further TSM documentation: http://www.ibm.com/software/tivoli

9.2 Tivoli Storage Manager and Content Manager

Content Manager, by itself, is only able to store data onto fixed disks attached to Resource Managers. If further storage options are required, such as the use of LTO tape, Tivoli Storage Manager should be considered. It should be noted that TSM is the hierarchical storage management (HSM) product that Content Manager on Windows or UNIX supports.

The TSM program product is included with Content Manager. A limited-use TSM license is provided with Content Manager. This means that while there are no restrictions on the use of TSM with Content Manager such as backup and archive, the limited-licensed TSM can only be used with Content Manager. You should not back up other systems that are not a part of Content Manager using this license.

Content Manager provides out-of-the-box integration with Tivoli Storage Manager. Integrating Content Manager to store data to TSM is easy to implement, with no application development required. When used in conjunction with TSM, Content Manager utilizes many of the storage benefits of TSM such as the built in TSM feature that enables you to take an extra copy of archived data via copy storage pools.
With the integration of TSM, Content Manager is able to support writing to WORM media (optical and tape) for permanent, unalterable storage. With the integration, TSM can write multiple copies of Content Manager data, and permits the storage of a copy at an offsite location.

Even though fixed disk costs are falling, it can still prove to be less expensive to use sequential access media to store large volumes of data in certain circumstances. Large amounts of data are what TSM is designed to manage. TSM provides the maximum storage flexibility for Content Manager.

For example, if you need to store your data for seven years for legal reasons and after the first year there is very little demand for retrieving or working with the data, it may be wise to consider a slower less expensive form of media for the last six years of the data’s storage life. This slower, less expensive form of media may be a high capacity LTO tape library, where 200 GB of data can be stored on each tape volume.

With TSM managing an automated LTO tape library, the TSM policies can be created so that after six years storage on tape, the data is either deleted or moved to a storage pool that has been specifically designed for off-site storage. If the data is deleted, TSM manages the newly created free space within the tapes and can reclaim this free space for use later, by newly backed up data. The first year of object storage can be handled by Content Manager, which uses fixed disks attached to the Resource Managers, or TSM itself can manage the fixed disk by utilizing disk storage pools. You may want TSM to manage all storage, if, for instance, there is a lack of free fixed disk space directly attached to your Resource Managers.

Another advantage of having TSM manage object migrations is that it takes load off the Content Manager server. For example, if Content Manager stores objects for the first 30 days, and then migrates the data to a TSM disk pool after this time period is up, TSM can handle migrations to slower forms of media itself. This approach can be beneficial as it takes more time to migrate data from disk to tape/optical than to migrate data from disk to disk. If Content Manager is only performing disk to disk object migration and letting the TSM server do the time consuming object migration to slower forms of media, the Content Manager server allocates fewer resources to the migration task.

TSM supports hundreds of hardware devices, giving you a vast range of choice when deciding upon storage hardware and media for Content Manager.

**Note:** For the most recent list of hardware devices supported by TSM, refer to the following Web site:

**TSM levels supported by Content Manager**

TSM, Version 4.2.1 or later, is required for use with Content Manager, Version 8. This applies to both the TSM client API and the TSM server versions. At the time of the writing, Content Manager V8.2 is currently being shipped with TSM V5.1.5.

**Important:** Tivoli Storage Manager V5.1 (64 bit API client) on UNIX does not work with the Resource Manager.

Work-around: To enable the Resource Manager to use the 64 bit TSM API client on UNIX, copy the ..../tivoli/tsm/client/api/bin64/libApiTSM64.a file to libApiDS.a in the same directory

**Configuration requirements**

The TSM server can reside on the same machine (or node) as the Resource Manager. This improves communications performance, but divides processor and other system resources, between both servers. If the TSM server is installed on a different machine, the Resource Manager is able to interact with a TSM server on any TSM-supported platform.

Regardless of whether the TSM server is installed on the same physical machine or a different physical machine to the Resource Manager, the TSM client API is required for communication. This is because the Resource Manager uses the local TSM client API to store objects into the TSM server. The TSM server is managed and administered independently of the Resource Manager.

Before Content Manager is configured to store data to TSM, the following conditions must be met:

- The required TSM policies, management classes, storage pools, and volumes are defined accordingly.
- The required TSM storage pools and volumes are online.
- The TSM storage pools and volumes that are to be used by the Resource Managers have sufficient storage space.
- The TSM server needs to be active when the Resource Manager needs to read from or write to its storage repository.

Before Content Manager can begin to use TSM for storage purposes, there are many simple configuration steps that you need to perform on both the Resource Manager and the TSM server.
9.2.1 TSM server configuration

Before a Resource Manager can be configured to utilize TSM, there are a number of definition commands that must be executed on the TSM server. These definition commands can be entered by any of the administrative interfaces for the TSM server. The following sequence of definitions are provided in the TSM administrator command format:

1. DEFINE DOMAIN (define a new policy domain)
2. DEFINE POLICYSET (define a new policy set)
3. DEFINE MGMTCLASS (define a new management class)
4. DEFINE COPYGROUP (define a new backup copy group)
5. ASSIGN DEFMGMTCLASS (assign a default management class)
6. VALIDATE POLICYSET (verify a policy set)
7. VALIDATE POLICYSET (verify a policy set)
8. REGISTER NODE (register a client node)

The commands listed above are in addition to commands that must be executed in order to define the physical storage device to TSM (such as an automated tape library) and to label the media to be written to. For a complete set of these commands and information on the definition commands listed above, refer to the IBM Tivoli Storage Manager Administrator’s Reference, GC32-0769.

9.2.2 Customizing the TSM API client files

When the prerequisite TSM API client files are installed onto the Resource Manager, a TSM API client options file is installed (called dsm.opt.smp). The TSM API client option file contains the configuration information needed by the TSM Client APIs to access TSM servers. The TSM server name, its port, the protocol to be used to communicate with the server, and the node name to use when connecting to the TSM server are only a few examples of what is contained in the client option file.

Your workstation can contain more than one client option file. For example, in some situations, a workstation may have both the Resource Manager server and the TSM Backup Archive Client installed on it. The Resource Manager is most likely configured to communicate with a TSM server dedicated for Content Manager, while the TSM Backup Archive Client is configured to back up workstation files to the company’s central TSM server. In this circumstance, you can create a client option file tailored for the Resource Manager and another client option file tailored for the TSM backup client.
The default name of the file is dsm.opt; it is the usual practice to copy dsm.opt.smp and rename the copy to dsm.opt and then edit this file. Here are some points to consider when editing the options file:

- For performance and reliability reasons, you should configure the Resource Manager to use the TSM API password access PROMPT.
- The TSM API access method GENERATE is supported, but the Resource Manager first attempts to access TSM with PROMPT. If the PROMPT password access method is not successful, it retries, using GENERATE.
- Be careful not to make any typing errors while editing the options file, as any syntax error invalidates the entire client options file, preventing Content Manager from migrating objects to, or retrieving objects from, TSM.

**Important:** Tivoli Storage Manager API client (UNIX) password access GENERATE without TCA is not supported

Each Resource Manager needs to have a TSM API client options file configured locally, and an ICMRM.properties file. If the Resource Manager is installed within a UNIX environment, there is further client file to configure, called dsm.sys.

For further information on configuring TSM options files and sample client options files tailored for a CM/TSM integration, refer to:

- *IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332

### 9.2.3 Configuring a Resource Manager to use TSM

Once the TSM client API options file (and dsm.sys file on UNIX) has been configured on each Resource Manager, the Resource Manager now must be configured for TSM use.

The ICMRM.properties file must be updated to reflect:

- The location of the TSM client options file created earlier - DSMI_CONFIG
- The path to the TSM message file, dscameng - DSMI_DIR
- The path to the TSM client API log file - DSMI_LOG
By default, the ICMRM.properties file is found within the path where you chose to install WebSphere on your Resource Manager. For example, this is the location of ICMRM.properties file:

```
C:\WebSphere\AppServer\installedApps\IM01\icmrm.ear\icmrm.war\WEB-INF\classes\com\ibm\mm\icmrm\ICMRM.properties
```

After updating the ICMRM.properties file, the following steps should be performed using the Content Manager System Administration Client:

1. Define a new server.
2. Define a new storage class.
3. Define a new TSM volume in the storage systems.
4. Enable the TSM device manager (ICMADDM).

Once all of the steps are completed, Content Manager can now use TSM for object storage.

Each TSM volume defined for the Resource Manager results in a unique TSM file space on the TSM server.

The name of the file space is:

```
/ICM/resource-manager-name/resource-manager-collection/TSM-management-class
```

When the first object is stored into each unique Content Manager TSM volume, a TSM file space is created.

When all of the objects are deleted or migrated out of the TSM file space, the initial file space is not deleted.

To delete an empty file space, the TSM administration function should be used.

**Important:** Content Manager does not check for a full TSM storage pool. The defined Content Manager volume pointing to TSM is considered to be infinite in size. The TSM system administrator is responsible for ensuring that all the storage pool volumes associated with the target management class are online, and have sufficient storage space for backing up objects.
**Using overflow storage systems**

When creating volumes, there is an option to mark them as overflow volumes. This means that the volume becomes an overflow storage area for all storage groups. Overflow volumes store objects when all other volumes for a storage group are full, unless one of the volumes within the group is a TSM volume, as these are never considered *full* by Content Manager.

If a storage class has both file systems (AIX) or volumes (Windows) and TSM storage systems assigned to a storage group, the file system or volume is used for storing objects first. When all of the assigned file systems or volumes are full, objects are stored to TSM.

If a storage class has both a file system or volume and a TSM storage system marked as overflow storage systems, the first available overflow storage system, based upon its creation date, is used when all the assigned storage systems are full.

When the first object is stored to a storage system that is marked as overflow, the storage system is assigned to the storage group to which the object belonged.

Because TSM acts as an infinite object storage repository, the concept of using a TSM management class as an overflow storage system is different than using a volume or file system as an overflow storage system.

For further information on the steps needed to integrate Content Manager with TSM and configuring the Content Manager System Administration Client, refer to:

- *IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332
- *IBM Content Manager for Multiplatforms - System Administration Guide*, SC27-1335
XML support

This chapter provides an overview of the new Content Manager V8.3 XML services such as the Content Manager XML schema mapping tool, runtime APIs, and Web services.

We take a look at what services can do and how they might be used within an application to support business processes based on content.
10.1 Introduction

This chapter covers XML support in Content Manager. It should be used in conjunction with the following IBM product manuals:

- *IBM DB2 Content Manager for Multiplatforms System Administration Guide*, SZ27-1335

XML or Extensible Markup Language is a markup language that you can use to create your own tags. It was created by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis of all Web Pages.

XML, similar to an extensible tag language, can describe complicated structures in ways that are easy for programs, and people, to understand. Web services depend heavily on XML. XML uses textual data instead of binary data to represent data. For example, integers are often represented differently by the various hardware and software programming languages being used today; however, the textual way that XML represents data (in this case integer) makes it language and platform independent. This independence save you time and resource when integrating your applications with Content Manager.

With Content Manager V8.3, several XML services can be supported. For example, XML messages in a SOAP envelope can be accepted using the Content Manager HTTP interface to execute tasks such as import, export, search, create, update, retrieve, delete and document routing.

10.2 How XML services work with other Content Manager programming layers

Certain programming layers require converted XML objects to work with the Content Manager connector. The layers include:

- Web services: Content Manager HTTP interface that accepts your XML messages (defined by cmbmessages.xsd) in a SOAP envelope to perform runtime operations such as import, export, search, create, update, retrieve, delete, and document routing. The Web services automatically wrap and extract the XML messages in the SOAP message, and send them to the XML messaging JavaBean.

- JavaBeans™ (XML): Reusable Java classes based on the Content Manager connector XML APIs and the Information Integrator for Content JavaBeans.
The XML JavaBeans perform runtime operations such as import, export, search, create, update, retrieve, delete, and document routing. In particular, the CMBXMLMessage bean parses all XML messages based on the cmbmessages.xsd schema.

- **XML schema mapping utility (XML)**: XML conversion tool that can convert a user-defined schema into the storage schema that Content Manager supports.

**Important**: In this context, storage schema is essentially the representation of the item type in XML schema form.

- **Content Manager connector (XML)**: XML application programming interfaces that can import and export data model metadata objects, administrative metadata objects, and data instance objects.

We do not cover the XML export/import tool in this chapter. It is covered in detail in **Chapter 19, “Export and import utilities” on page 509**.

Figure 10-1 shows how Content Manager V8.3 XML layers relate to the Content Manager Connector.

![Figure 10-1 XML Services programming layers](image)

<table>
<thead>
<tr>
<th>System administration client (XML)</th>
<th>Web Services</th>
<th>eClient</th>
<th>XML Schema Mapping Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JavaBeans (XML)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**10.3 Working with Web services**

Content Manager provides a self-contained, self-describing modular interface, called the Web services interface, that you can use within your applications, with other Web services interfaces, or in complex business processes to seamlessly perform actions in a Content Manager system. A Web service interface is a reusable, loosely coupled, software component that can be located, published and invoked through a network, such as the Web.
With the Web services interface, you can dynamically integrate your applications with Content Manager, regardless of the programming language that they were written in and the operating system that they run on. You can use the Web services interface to do something as simple as viewing a text document or you can incorporate the Web services interface into more complex business applications or processes.

For example, in an insurance scenario, you can incorporate a Web services interface into an existing Web application that allows your customers to print their current auto policy. Furthermore, you can incorporate another Web services interface into the same application that allows your customers to view the current Blue Book value of their car.

Web services is a standards-based mechanism for accessing a system using XML-based messaging over a messaging bus based on the HTTP protocol. Content Manager V8.3 delivers a set of out-of-the-box Web services operations. The Content Manager Web services support leverages the power of WebSphere Application Server and the Content Manager Java stack to support remote access to Content Manager functionality and semantics. Using the Web services interface lets users integrate applications with Content Manager regardless of the programming language the applications were written in or the platform they run on. Some key features are:

- Support for .NET client applications including support for DIME attachments
- Support for Java and J2EE client applications including support for MIME attachments
- Support for automatically generating WSDLs specific to your system
- Support for all core content and document routing functions

### 10.3.1 Web services overview

Web services is an emerging technology that is becoming the technology of choice for application integration. The key attribute of Web services is that they define a program-to-program, services-oriented communications model that is based on an XML messaging format. The Web services model is built on existing and emerging standards, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Hyper Text Transfer Protocol (HTTP), and the Web Services Description Language (WSDL).

Simple Access Object Protocol (SOAP) is an XML-based messaging protocol that is used as the basis for Web services interactions between two applications. All Web services communication is done by using SOAP messages. A SOAP message contains the following elements:
- Envelope
- Header (optional)
- Body
- Attachments (optional)

Typically, a SOAP envelope, with zero or more attachments, represents a SOAP message. The SOAP message envelope contains the header and the body of the message. The SOAP message attachments enable the message to contain data, which can include XML and non-XML data (such as text and binary files). SOAP headers are used to describe the context and the purpose of the message. SOAP headers also provide mechanisms to extend a SOAP message for adding features and defining high-level functionality such as security, priority, and auditing.

SOAP allows you to invoke Web services in two ways: RPC (Remote Procedure Call) messaging and document-style messaging. The Content Manager Web services use the document-style method for invoking Web services because it is much more flexible than the RPC method.

In a service-oriented architecture, the interface definition is crucial. It is the interface definition that serves as the contract between what the Web service provides and what the client can expect. Web services use WSDL, another set of XML tags that are used to describe the Web services interface. The types of things that WSDL describes are the location of the Web service, how to connect to it, which parameters must be passed in the SOAP request, and which values to return. The WSDL also provides binding information.

The Web services model leverages the XML, HTTP, SOAP, and WSDL technologies and protocols to provide an environment that makes application integration easier, faster, and more cost effective. Web services allow any network-enabled, XML-aware application to invoke a Web service regardless of the programming language or operating system involved.

Web services provide the following advantages:

- **Flexibility.** Universal interfaces do not have to change with the inevitable software changes that are caused by changing business needs.
- **Agility and productivity.** Rapid application assembly tools allow you to quickly integrate Web services into new business processes or experiment with new business ideas.
- **Cost savings.** Reduce staffing requirements, replace paper processing, reduce errors.
- **Leverage existing investments.** You can use old software in new ways by building a Web services layer for universal access.
Runtime Web services include:
- AddItemToFolderRequest
- ChangePasswordRequest
- CheckInItemRequest
- CheckOutItemRequest
- CreateItemRequest
- CreateLinkRequest
- DeleteLinkRequest
- DeleteItemRequest
- GetPrivilegesRequest
- ListSchemaRequest
- ListServersRequest
- MoveItemRequest
- RemoveItemFromFolderRequest
- RetrieveFolderItemsRequest
- RetrieveFoldersForItemRequest
- RetrieveItemRequest
- RunQueryRequest
- UpdateItemRequest

Document Routing Web services include:
- ContinueProcessRequest
- ListActionRequest
- ListActionNamesRequest
- ListNextWorkPackageRequest
- ListProcessRequest
- ListProcessesRequest
- ListProcessNamesRequest
- ListWorkListNamesRequest
- ListWorkListsRequest
- ListWorkNodeRequest
- ListWorkNodesRequest
- ListWorkPackageCheckOutOptionRequest
- ListWorkPackagesRequest
- ResumeProcessRequest
- SuspendProcessRequest
- TerminateProcessRequest
- UpdateWorkPackageRequest

To work with the Content Manager Web services, you must have a working knowledge about them. To find out more about the Web services standards, see the World Wide Web Consortium (W3C) Web site:

http://www.w3.org
10.3.2 Content Manager Web services implementation

The Content Manager Web services interface architecture is based on a messaging communications model, in which the entire documents are exchanged between service clients and servers. This messaging based model provides several benefits.

One benefit of the document messaging based model is that the XML specification was developed to allow ordinary data, that is usually locked up in a proprietary format, to be described in an open format that is readable by humans, self-describing, and self-validating. When a Web service uses document messaging, it can use the full capabilities of XML to describe and validate a high-level business document. Another benefit is that even though enhancements and changes are made to the XML schema, the calling application will not break.

Lastly, the document messaging model makes object exchange more flexible, because the design of a business document is often well suited to object-oriented architectures. As a result, two applications can be designed to exchange the state of an object by using XML. In contrast with object serialization, in an object exchange each end of the exchange is free to design the object as it sees fit as long as the exchange conforms to the agreed upon XML document format. One reason for not using object serialization is to support client-side and server-side implementations of an object. Many current industry-specific XML schemas are designed as client-server architectures in which the processing that is done at the client is separate from the processing intended at the server. As is often the case, the client is simply requesting or saving information in a specific document format that is persists on the server.

The main components in the Content Manager Web services model include the requester, the Web services server, the XML beans layer, and the Content Manager repository. They interact in the following steps:

1. A requester makes a call to the Web services server.
2. The Content Manager Web services server analyzes and extracts the XML message from the SOAP envelope.
3. The XML message is sent to the Content Manager XML beans layer.
4. The XML beans transform the XML into multiple calls to the underlying Content Manager APIs.
5. The APIs access the data in the repository and return values to the XML beans.
6. The return values from the APIs are transformed into an XML response message by the XML bean. This message contains the request status, response data and attachments, and exception information, if applicable.
7. The message is returned to the Content Manager Web services server.
8. The Web services server creates a SOAP message, which can include attachments, from the response data and returns the message to the requester.

Figure 10-2 depicts the steps involved in document processing using Web services. A SOAP request is sent to the Web services server to store an insurance claim. The Web server processes the SOAP request and sends the data to the server. The claim is stored into the Library Server and pictures associated with the claim are stored in the Resource Manager.

![Figure 10-2 Processing a document using Web services](image)

**Important:** If you upgrade the Content Manager Web services application from WebSphere Application Server Version 5 to WebSphere Application Server Version 6 using the WebSphere Application Server tools, Content Manager Web services will not work properly. You must install Content Manager Version 8.3 Fix Pack 1 after you upgrade to WebSphere Application Server Version 6.

### 10.3.3 Integrating basic Web services into your applications or processes

This section explains how to develop client applications in order to interact with the Content Manager V8.3 Web services interface. You can communicate with the interface through the Web services Description Language (WSDL) in two ways, described in the following sections:
**WSDL generation**

You can write an application that uses a WSDL utility to automatically handle the XML/SOAP requests and responses based on the structure of your item types.

The sample classes for this application are written in C#. This requires the Web service support and wsdl.exe utility provided in Microsoft Visual Studio® .NET 2003. All C# sample classes are located in:

IBMCMROOT/samples/webservices/CMWebServiceClient

For .NET clients, the WSDL does not describe the syntax of the input and output of the messages. They are defined as xs:AnyType. You should use the XML beans messages schema file (cmbmessages.xsd) and the item type schema files to generate the XML request documents and send them to the Web services using the URL specified in the WSDL file.

Toolkits such as the Microsoft SOAP Toolkit, provide low-level APIs for generating and exchanging SOAP messages. These APIs allow you to specify an XML document that represents the body of a SOAP message and sends the document to the Web service URL and returns the reply document as part of the SOAP message. This is a low-level interaction with the Web services. This type of interaction allows the most flexibility because the Web services interface does not change, even if the XML schema changes.

The disadvantage to using the Web service in this manner is that there is the burden, on the development side, of generating XML messages and dealing with low-level APIs for sending and receiving SOAP messages. In this case, the WSDL is used only for specifying the end point URL of the Web service.

To create XML/SOAP requests, you can write a CMWebServiceClient application that utilizes a WSDL utility. A WSDL utility can automatically process your XML/SOAP requests and responses by representing them as proxy classes. For example, Microsoft provides a wsdl.exe utility that can represent XML documents as C# proxy classes.

To write an application that interfaces with Microsoft .NET's WSDL utility, follow these steps:

1. Install the following software:
   a. Content Manager V8.3 Web services toolkit
   b. Microsoft Visual Studio .NET 2003
   c. .NET Framework SDK Version 1.1 from the Microsoft Web site.
   d. .NET Web Service Enhancements SP1 from the Microsoft Web site.
2. Load the First Steps XYZ insurance samples.
3. In the following file:
```
%IBMCMROOT%\samples\webservices\CMWebServiceClient\CMWebService.cs
```
Replace all instances of localhost with the host name of your Web services server.

4. Load the CMWebServiceClient.csproj into Visual Studio .NET.

5. Program the Web services application using CMWebServiceClient.cs as guidance. For details about programming Web services requests in C#.

6. Run the sample by entering the command:
```
CMWebServiceClient.exe icmn1sdb icmadmin password
```
Where icmn1sdb represents the Content Manager server, icmadmin represents your system administration ID, and password represents the password.

**Important:** Note that the Web services samples assume the server is local. However, the server and the Web services samples can be installed on different machines, in which case you must modify the server URL accordingly.

**XML/SOAP requests**
You can write an application to send your own XML requests through a SOAP envelope to the Web services server. This server translates the request into calls on the XML Handler bean, and then sends XML/SOAP responses back to your application.

The sample classes for this application are written in Java, which requires a Web services JAR file from the WebSphere Studio Version 5.1 or Version 6.0 Web services toolkit. All sample Java classes are located in
```
IBMCMROOT/samples/webservices/GenericWebServiceSample
```
You can customize your own Java client to create the XML requests to send to Web services. For Java clients, there is one operation called `processXMLRequest`, which describes two input parameters.

The first parameter is an XML string that represents the XML request for the Web services. The second parameter is a `javax.mail.internet.MimeMultipart` object, which represents the attachment representing a document or resource object. You must generate this XML string using the XML Beans messages schema file (`cmbmessage.xsd`) and the item type schema files.
You can use any JAX-RPC based client toolkit to generate the classes that will invoke the Web services and pass the parameters back and forth to the Web services. WebSphere Application Server Version 5.1 provides a client-side tool called WSDL2Java that you can use to generate the client-side classes for the Web services. Because the WSDL file does not define the syntax of the XML documents, the interface of the Web services does not change if the XML schema for the request changes.

To write an application in the Java environment, follow these steps:

1. Install the following software:
   a. Content Manager Version 8 Release 3 Web services toolkit
   b. WebSphere Application Server Version 5.1

2. Load the First Steps XYZ insurance samples.

3. In the following file:
   
   %IBMCMROOT%/samples/webservices/GenericWebServiceSample/sample/
   CMBGenericWebServiceServiceLocator.java

   Modify the CMBGenericWebService_address variable with the name of your Web services server rather than localhost.

4. Program the Web services application using GenericWebServiceSample.java as guidance.

5. Compile the proxy classes and GenericWebServiceSample.java with the CLASSPATH with the following WebSphere JAR files: activation.jar, j2ee.jar, mail.jar, qname.jar, webservices.jar, and wsdl4j.jar.

6. Run the sample by entering the command:
   
   java GenericWebServiceSample icmn1sdb icmadmin password

   Where icmn1sdb represents the Content Manager server, icmadmin represents your system administration ID, and password represents the password.

**WebSphere Studio Application Developer (WSAD) only**

If you are creating a Web services client using WebSphere Studio Application Developer (WSAD) Version 5.1, then within that wizard you must select Define custom mapping for namespace to package, and specify a different package for each of the namespaces:

http://www.ibm.com/xmlns/db2/cm/api/1.0/schema
http://www.ibm.com/xmlns/db2/cm/beans/1.0/schema
http://www.ibm.com/xmlns/db2/cm/webservices/1.0/schema

Also needed is a package for no namespace, which you can specify as an empty string.
**Business Process Choreographer only**

If you using the generated WSDL from the system administration client for Business Process Choreographer, you must edit the cmbmessages_modified.xsd file (which is located in the ZIP file along the WSDL file), and change the line

From:

```xml
<xs:import xmlns="http://www.w3.org/2001/XMLSchema"
schemaLocation="itemtype_modified.xsd"/>
```

To:

```xml
<xs:include xmlns="http://www.w3.org/2001/XMLSchema"
schemaLocation="itemtype_modified.xsd"/>
```

### 10.3.4 Exporting item types to a WSDL file

You can export your item types into Web Services Description Language (WSDL) files.

DB2 Content Manager provides a self-contained, self-describing modular interface, called a Web services interface, that you can use within your applications, with other Web services interfaces, or in complex business processes to seamlessly access items stored in a DB2 Content Manager system. A Web service interface is a reusable, loosely coupled, software component that can be located, published and invoked through a network, like the Web. The Web services model leverages the WSDL and other technologies and protocols to provide an environment that makes application integration easier, faster and more cost effective.

To export your item type to a WSDL file, use these steps:

1. From the System Administration Client window, right-click an item type and select **Export to WSDL file** (Figure 17-6) to open the Save WSDL File As window (Figure 17-7).
2. Browse to the directory where you want to store the file (Figure 17-7).
3. Enter the name of the file.
4. Click **Save**.

---

Figure 10-3  Export Item Type to WSDL file, Step 1

Figure 10-4  Save WSDL File As window
How the Web services work with development toolkits

The DB2 Content Manager Web service is a messaging-based communication model that defines loosely coupled and document-driven communication. The client service requester invokes the Web service by sending it a complete XML document that represents a particular request for a DB2 Content Manager operation, such as search. The DB2 Content Manager Web service provider receives the document, processes it, and returns a message, as an XML document.

When you install of the DB2 Content Manager Web service, two WSDL locations that describe the operations and end points of the Web service are provided by DB2 Content Manager. Your application environment is the determining factor for choosing which WSDL location to use.

There are a number of Web services toolkits that can take a WSDL file and create a set of classes to client-side representation of the Web service, the request, and any reply messages. The benefit of using development toolkits is that you do not have to create the XML document yourself because toolkits can create classes that generate the XML request for you. The toolkit serializes the classes into XML and creates and exchanges the SOAP messages with the Web service. This makes client-side development much easier and faster.

In order for a tool to create the classes, the WSDL must thoroughly describe the syntax of the input and output messages, as well as the operations. Because the schema of the user-defined item types are not necessarily known at installation time, you must create the WSDL for the item types after you have completely installed DB2 Content Manager.

You can generated a WSDL for any item type using the DB2 Content Manager System Administration Client, and use that WSDL to perform operations provided by the Web service.

Tools such as the WSDL.exe provided by the Microsoft .NET Framework SDK or the Web Reference feature in Microsoft Visual Studio .NET can take a WSDL file and create a set of classes that you can use to invoke the DB2 Content Manager Web services. The WSDLs generated by the DB2 Content Manager System Administration Client support clients built in .NET and Java.
Figure 17-8 illustrates the process of creating WSDLs for use in a .NET environment.

![Diagram of XML support process]

**Figure 10-5  Tooling support for use in a .NET environment**

### 10.3.5 Exporting a process as XML Text (Workflow)

Before you can export a process as XML, you must create and verify it in the graphical process builder. The verification process does not have to complete successfully before you can export the process. To export a previously created and verified process, you must first open it in the graphical process builder.

You can export a new process that you have open in the builder or an existing process. The primary reason to use this functionality is to move built and verified processes from a test system to a separate production system.

Do not confuse this functionality with the XML export functionality that is available from the System Administration Client window - that XML export function exports a full range of system administration data as binary XML, this XML export function exports only the content of the graphical builder as XML text for import within the graphical builder on another system.

Although you are exporting the content of the graphical builder, you are not exporting the definition of the included document routing objects (for example, work nodes). If the necessary document routing objects do not exist on the target system, you can use the XML export functionality from the System Administration Client window to export them.
To export a process from the graphical builder as XML text:

1. Within the graphical process builder, click **File → Export XML text** (Figure 17-9).

![Export XML text](image1)

*Figure 10-6  Export XML text*

2. Specify a name and location for the exported XML file (Figure 17-10).

![Specifying a name and location for the XML file](image2)

*Figure 10-7  Specifying a name and location for the XML file*

3. Click **Save XML File**.

The file is exported as a XML text file.
10.4 XML JavaBeans

The XML JavaBeans are Java classes that provide convenient interfaces to the Content Manager connector XML APIs and the Information Integrator for Content JavaBeans. They also serve as the communication layer between the Web services and the connector APIs.

The XML JavaBeans can perform run-time operations such as import, export, search, create, update, retrieve, delete and document routing. They do not support system administration functions.

If you decide to program applications that communicate with Content Manager directly through the XML JavaBeans, you can direct your XML request straight to the CMBXMLMessage bean (similar to what the Web services do). Your XML requests must follow the structure described in the cmbmessages.xsd schema.

Example 10-1 is a JavaBean example that sets up a CMBXMLMessage bean to send an XML search request directly to it.

Example 10-1 XML JavaBean example Part 1

```java
public class TXMLSearch2 {
    public static void main(String[] args) throws Exception {
        // Create beans
        CMBXMLServices xmlServices = new CMBXMLServices();
        // Create the search request message and get the reply message
        CMBXMLMessage reply = search(xmlServices, dstype, server, userid, password, entity, condition);
        System.out.println("Search reply: " + reply.getAsString());
    }

    static public CMBXMLMessage search(CMBXMLServices xmlServices, String dstype, String server, String userid, String password, String entity, String condition) throws CMBException, Exception {
        return search(xmlServices, dstype, server, userid, password, entity, condition, null);
    }

    static public CMBXMLMessage search(CMBXMLServices xmlServices, String dstype, String server, String userid, String password, String entity, String condition, String maxResults) throws CMBException, Exception {
        // Create the query string
        int queryType = CMBBaseConstant.CMB_QS_TYPE_XPATH;
        String queryString = "/" + entity;
        queryString += "]" + condition + "];
        String connectString = "";
```
/ If the server name is followed by a parenthesized string,
// use that string for the connect string.
// e.g. ICMNLSDB(SCHEMA=ICMADMIN)
if (server.indexOf("\") > 0) {
    connectString = server.substring(server.indexOf("\") + 1);
    server = server.substring(0, server.indexOf("\") + 1);
    if (connectString.endsWith("\") ) {
        connectString = connectString.substring(0,
                connectString.length() - 1);
    }
}
// continued...

To send an XML search request using the above example (Example 10-1),
you can pass in your server name (ICMNLSDB in the example)
and connectString as SCHEMA=ICMADMIN. See Example 10-2.

Example 10-2  XML JavaBean example Part 2

StringBuffer XMLBuffer = new StringBuffer();
String maxResString = "";
if (maxResults != null) maxResString="maxResults=\\" +
maxResults + "\\"";
XMLBuffer.append("<?xml version="1.0" encoding="UTF-8" ?>");
XMLBuffer.append("<RunQueryRequest " + maxResString +
" version=" + CMBXMLConstant.CMB_LATEST_VERSION + "
" retrieveOption="" + CMBXMLConstant.CMB_RETRIEVE_CONTENT +
" contentOption="" + CMBXMLConstant.CMB_CONTENT_ATTACHMENTS +
" " + TXMLTestcase.namespace + ">"));
XMLBuffer.append("<AuthenticationData connectString="" +
connectString + "\" configString="" >");
    XMLBuffer.append("<ServerDef>");
    XMLBuffer.append("<ServerType>" + dstype + "</ServerType>");
    XMLBuffer.append("<ServerName>" + server + "</ServerName>");
XMLBuffer.append("</ServerDef>");
XMLBuffer.append("<LoginData>");
XMLBuffer.append("<UserID>" + userid + "</UserID>");
XMLBuffer.append("<Password>" + password + "</Password>");
XMLBuffer.append("</LoginData>");
    XMLBuffer.append("</AuthenticationData>");
XMLBuffer.append("<QueryCriteria>");
XMLBuffer.append("<QueryString>" + queryString + "</QueryString>");
XMLBuffer.append("</QueryCriteria>");
XMLBuffer.append("<RunQueryRequest>");
System.out.println("XMLRequest: 

" + XMLBuffer.toString()+ 

");
CMBXMLMessage doc = new CMBXMLMessage(XMLBuffer.toString(), null);
// Search using the makeRequest method on the CMBXMLServices bean
System.out.println("Performing search");
System.out.println(XMLBuffer.toString());
```java
CMBXMLMessage reply = xmlServices.makeRequest(doc);
System.out.println("Search reply");
System.out.println(reply.getAsString());
TXMLTestcase.printAttachments(reply.getAttachments());
return reply;
}
}
```

**XML services bean and helper classes**

There are other kind of JavaBeans which belongs to Information Integrator for Content. The Information Integrator for Content provides two types of JavaBeans:

- **Non-visual beans**: Use the non-visual beans to build Java and Web client applications that require a customized user interface. The non-visual beans support the standard bean programming model by providing default constructors, properties, events and serializable interface. You can use the non-visual beans in builder tools that support introspection.

- **Visual beans**: The visual beans are customizable, Swing-based, graphical user interface components. Use the visual beans to build Java applications for Windows. You can place them in conjunction with the non-visual beans when building an application.

The XML services bean and helper classes are part of the non-visual beans provided by Information Integrator for Content.

The XML bean takes messages in the form of XML requests and replies to perform various operations on Content Manager, including searching, creating, updating, exporting items into XML format, export item type definitions as XML schemas, and workflow operations. The XML bean leverages the XML support in the API for item import, item export, and schema export operations.

The XML services bean and helper classes include:

- **CMBXMLServices**: Constructor for the XML services bean.
- **CMBXMLMessage**: Used as a wrapper class for an XML document to describe a request or reply to the beans. It contains both the XML source of the request document, and the set of attachments associated with the message. The XML document may be a file, string, input stream, or document object model (DOM).
- **CMBXMLAttachment**: Used together with the CMBXMLMessage class to represent an attachment in the XML message. This object maps to a resource object or a document part.
10.5 XML schema mapping utility

Content Manager provides both a graphical interface and APIs to convert a user-defined schema into a storage schema that can be imported into the system. The tool can also generate an XSLT (the Extensible Style sheet Language Transformations) query script. It can be saved as part of a mapping in a repository. Using this script, you can program an application that automatically converts XML documents from the user-defined schema to the storage schema.

A new XML schema mapping tool simplifies the process of defining the Content Manager schema to support incoming XML documents adhering to specific XML schemas. The tool dynamically maps an XML schema to a Content Manager schema either automatically (including creating the new Content Manager schema) or manually through the use of the graphical XML schema mapping utility. Once the mapping is generated, XML documents adhering to the mapped XML schema may be captured, shredded, stored and managed in Content Manager automatically, with no user interaction.

The XML schema mapping tool enables you to convert an XML schema file (.xsd) into another XML schema that you can import as a Content Manager item type. The source schema can be automatically transformed into a default storage schema. You can edit the structure and some properties of the storage schema using the XML schema mapping tool before you import it into Content Manager. In addition, you can export an existing Content Manager item type as a storage schema and directly enter a mapping between a source schema and the item type. Use of the XML schema mapping utility results in the automatic creation of an additional item type and set of attributes. These objects will display with other similar objects in the System Administration Client.

**Important:** There are some restrictions:
- DTD schemas are not supported. There are many programs available to convert DTDs into XML schemas.
- Annotation support is not complete in Content Manager V8.3 of the XML schema mapping tool. When a target schema is loaded, some annotations will be lost, and none will be preserved when saving a mapping.

The XML schema mapping tool keeps the mapping between the source and storage schema and creates an XSLT script that you can use later to transform XML documents that conform to the source schema into documents that conform to the storage schema. You can import these transformed XML documents into Content Manager as items. The XML schema mapping tool uses the working Content Manager server as persistent storage, enabling you to save and share your XML mappings.
10.5.1 Supported scenarios

The XML schema mapping tool supports the following scenarios when developing your schema mapping:

- Creating schema mappings with a brand-new storage schema
- Creating schema mappings with a pre-existing storage schema
- Revise existing schema mappings

**Creating schema mappings with a brand-new storage schema**
You can convert your user-defined schema to a brand-new storage schema, and you can modify both the storage and mappings. You can then create a new item type from the storage schema, assign a mapping name, and save the mapping in a repository.

**Creating schema mappings with a pre-existing storage schema**
You can convert your user-defined schema to a previously created storage schema by manually mapping the user schema to the storage schema. You can then invoke the tool function that will generate a new XSLT query script. You can then assign a mapping name and save the mapping in a repository.

**Revise existing schema mappings**
You can re-open a previously created mapping (using the mapping name) and modify both it and the storage schema. You can then save the modified storage schema, user-defined schema and new XSLT query script back to the repository.

10.5.2 Creating an XML schema file

When you open the tool, there are three main sections or panes, from left to right: the mapping navigator, the user schema viewer, and the storage schema viewer/editor. Table 10-1 describes the functions of each pane.

**Table 10-1 XML schema mapping tool panes**

<table>
<thead>
<tr>
<th>Pane name</th>
<th>Pane function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping pane</td>
<td>Display the current state of the mapping. A mapping is composed of the following parts:</td>
</tr>
<tr>
<td></td>
<td>- Mapping name</td>
</tr>
<tr>
<td></td>
<td>- Source schema</td>
</tr>
<tr>
<td></td>
<td>- Target schema</td>
</tr>
<tr>
<td></td>
<td>- A list of correspondences between the source and target schemas</td>
</tr>
<tr>
<td></td>
<td>- The queries generated by the correspondences</td>
</tr>
</tbody>
</table>
The process to create an input XML schema file from an existing XML schema file (.xsd) consists of the following tasks:

1. Create a mapping.
2. Select a source schema.
3. Generate the storage schema.

Each of these tasks is described in detail below.

**Creating a mapping**

To create a mapping, follow these steps:

1. Select **Start → Programs → IBM Content Manager Enterprise Edition → XML schema mapping tool**.
2. Select **Mapping → New** from the main menu or click the new mapping button on the toolbar. The tool creates a new, empty map and displays it in the mapping pane. The default name for this new mapping is New_Mapping1.
3. Rename the new mapping to something meaningful by clicking New_Mapping1 to select it, then right-clicking and selecting **Rename** from the context menu. The Rename Mapping window displays.
4. Enter a new name and click **OK**.

**Selecting a user schema**

Next, you select the XML schema to convert into a storage schema. Follow these steps:

1. In the mapping pane, click **Source Schema** to select it.
2. Right-click it and select **Add from file system** from the context menu. You can browse to the XSD file that you want to use. Any included or imported file from this schema is also loaded automatically.

The XML mapping tool requires one root element be identified for each schema loaded. If you have more than one root XML elements for a schema, you are prompted to select the name of the root element that you want to use from the Multiple Roots Detected window.

<table>
<thead>
<tr>
<th>Pane name</th>
<th>Pane function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source schema</td>
<td>Displays the user XML schema elements and attributes in tree form.</td>
</tr>
<tr>
<td>Target schema</td>
<td>Displays the storage XML schema elements and attributes, and enables you to modify the structure and properties of the storage schema.</td>
</tr>
</tbody>
</table>
The schema file name is added under Source Schema in the mapping pane, and the loaded schema appears in the source schema pane.

**Generating the storage schema**

To generate a storage schema from a given loaded source schema, follow these steps:

1. In the mapping pane, click **Target Schema** to select it.
2. Right-click it and select **Generate from Source Schema**.

The XML mapping tool analyzes the structure of the source schema and makes the necessary changes to create a valid storage schema. For example, all leaf-level elements are converted into XML schema attributes, string types are converted into variable length strings, and choice model groups are changed to sequences.

The tool keeps track of where each element and attribute in the source schema ends up in the storage schema.

### 10.5.3 Mapping a user-defined schema to a storage schema

Using the APIs provided with the XML schema mapping tool, map a user-defined schema to a storage schema.

As a first step in using the APIs, use methods in the DKSchemaConverter class to convert your XML schema (Example 10-3). The methods perform the following tasks:

- **convert()**: Converts the user-defined schema to a storage schema and optionally saves the mapping as an XSLT query script in a repository.
- **getStorageSchema()**: Retrieves the converted storage schema.
- **getXSLTQuery()**: Retrieves the XSLT query script that can automatically convert XML documents from the user-defined schema to the storage schema.

**Example 10-3 Using DKSchemaConverter class**

```java
import com.ibm.mm.sdk.common.DKException;
import com.ibm.mm.sdk.cs.DKDatastoreICM;
import com.ibm.mm.sdk.xml.schema.DKDocumentConverter;
import com.ibm.mm.sdk.xml.schema.DKMapperException;
...
DKDatastoreICM cmDatastore = new DKDatastoreICM();
    cmDatastore.connect(cmDatabase, cmUser, cmPassword, "");
    System.out.println("Connected.");
    File inputSchema = new File ( inputUserSchema );
    DKSchemaConverter converter = new DKSchemaConverter( cmDatastore );
```
if (mapName == null) {
    if (converter.convert( inputSchema.toURL(), rootElementName) == false) {
        System.err.println("dkConvert returned null.");
    }
} else {
    if (converter.convert( inputSchema.toURL(), rootElementName, mapName ) == false) {
        System.err.println("dkConvert returned null.");
    }
}

System.out.println("STORAGE SCHEMA:");
System.out.println( converter.getStorageSchema() );
System.out.println("XSLT Scripts");
String scripts[] = converter.getXSLTQuery();
System.out.println( scripts[0] );
System.out.println("-----------------------------");
System.out.println( scripts[1] );

As the second step in using the APIs, use methods in the DKDocumentConverter class to convert your XML documents (Example 10-4). The methods perform the following tasks:

- **getSchemaMappingNames().** Retrieves the schema mapping names from the repository.
- **getXSLTQuery().** Retrieves the XSLT query script that can automatically convert XML documents from the user-defined schema to the storage schema.
- **transformXMLDocument().** Transforms an XML document using the XSLT query script that you retrieved.
- **deleteSchemaMapping().** Deletes a schema mapping from the repository.

**Example 10-4  Using DKDocumentConverter class**

```java
import com.ibm.mm.sdk.common.DKException;
import com.ibm.mm.sdk.cs.DKDatastoreICM;
import com.ibm.mm.sdk.xml.schema.DKDocumentConverter;
import com.ibm.mm.sdk.xml.schema.DKMapperException;
...
DKDatastoreICM cmDatastore = new DKDatastoreICM();
    cmDatastore.connect(cmDatabase, cmUser, cmPassword, "");
System.out.println("MAPPING NAMES:");
Collection names=DKDocumentConverter.getSchemaMappingNames(cmDatastore);
System.out.println(names);
if (mapName == null)
```
return;
String[] query = DKDocumentConverter.getXSLTQuery(cmDatastore, mapName);
System.out.println("XSLT Scripts for " + mapName);
if (query == null)
    System.out.println("NONE.");
else {
    for (int i = 0; i < query.length; i++) {
        if (i > 0)
            System.out.println("----------------------------------");
        System.out.println(query[i]);
    }
}

if (inputXMLDoc == null)
    return;
File inputFile = new File( inputXMLDoc );
File outputFile = new File( "APIoutput.xml" );
DKDocumentConverter.transformXMLDocument( inputFile.toURL(),
    query, outputFile );
System.out.println("Output in APIoutput.xml");
Content Manager implementation

In this part of the book, we cover the Content Manager solution implementation process from planning and designing, to deployment. To put concepts into real practice, we provide a practical case study to demonstrate how to implement a Content Manager solution for a real-world scenario.

Starting from Content Manager V8.3, the installation and configuration has become much easier than in the previous version. This redbook will not address the installation and configuration for this version. Refer to the product manual for details. For reference purpose, we include the installation and configuration of Content Manager V8.2 in the appendix.
Planning and designing

Content Manager V8.3 has simplified the installation process and now includes a new product called Launch Pad, which provides links to tools for planning, documentation, and configuration scenarios.

In this chapter, we provide an overview of the planning steps required to successfully implement and migrate your solution with the IBM DB2 Content Manager. We cover a wide variety of topics such as business analysis, capacity planning, system configuration, and client options.
11.1 Planning basics

Planning is an important phase of any project. You may be in an organization implementing a Content Manager system for the first time, or in an organization that has disparate systems and trying to consolidate them into a single system. It is very important to plan ahead to minimize project risks and maximize the chances of a problem-free installation and implementation. The final goal is to have a quality system that meets your business requirements with efficiency.

11.2 Analyze business operations and requirements

An organization and its departments are intricately tied together with defined business processes to help meet organization goals. To successfully implement a Content Manager solution in your organization, the existing processes need to be understood clearly. Even if you know the basic processes already, it is essential that these are documented. Analysis often reveals more information than originally assumed. This includes how a specific business user actually accomplishes work or how certain documents interact with each other.

Implementing and migrating to a Content Manager solution is challenging. It may give you the opportunity to re-engineer the current practices in your organization. This may result in increased productivity, better return on investment, and greater service to customers, partners, and employees.

In order to document the current processes, you need to understand your organization’s content management goals and kick-start a successful Content Manager implementation. We recommend starting with the following information gathering template:

- What types of content need to be managed? Examples include invoices, sales orders, insurance policies, marketing brochures, and product manuals.
- What are the various electronic formats used and planned? Examples include Word, PDF, scanned images, and EDI documents.
- What is the expiration and retention policy of the content? Depending on legal and business requirements, this may be very important.
- What are the estimated sizes of the active electronic content? What is the projected growth rate over the next three years? For example, you have one terabyte of existing documents growing at 25% every year.
- Who will manage and use the content? Two additional questions arise:
  - What departments, divisions, or sub-organizations will manage or use the electronic content? Examples include Human Resource department, Public Affairs, Legal, Marketing, Products, and Customer Service.
Who are the content owners? Who are the users? What are their various roles? Examples include Authors, gate keepers, approvers, and reviewers.

What are the workflow requirements in requesting, creating, updating, deleting or archiving the content? Do they vary by business area, departments or some other classification? For example, documents made available to the Internet customers may have more approval steps to be completed by Legal gate keepers.

What are the migration requirements for migrating from existing systems into a Content Manager system? For example, you may need to migrate from a third-party vendor system to Content Manager, or from a Windows platform to an AIX platform.

How many users are expected in the final rollout of the system? For example, 4,000 users, including employees and some business partners, all need to access the system.

What are the accepted response time and throughput requirements during the peak periods of usage and during the normal business hours? What are the expected Business Volume Metrics (BVM)? For example, during the expected peak period between 8 AM and 9 AM, the system needs to handle 700 simultaneous users, with less than 5 seconds of response time for page launch expected. During the normal business hours, the response time requirement maybe less.

What is the existing infrastructure? It helps in understanding the infrastructure requirements for installing Content Manager. For example, you use the Windows platform, have an LDAP directory, and use Oracle or DB2 databases.

Will a custom data model be required? This depends on the items or the document types that are identified for your system.

What are the versioning requirements? For example, users need to retrieve up to 10 previous versions of the documents, or users need to retrieve documents modified up to 6 months in the past.

Are there any other considerations? Does your organization have any special usability requirements? For example, the system may need to be accessible to visually challenged users.

This is just a set of sample questions that can be used to start the analysis process and requirements interviews with the business areas. Each of the above questions can spawn to a separate design document into a typical Content Manager implementation project. Your IBM representative can help you with more detailed business analysis during the planning phase of the project.
Once all the information is collected and analyzed, you translate business requirements into the system requirements.

### 11.3 Planning and designing system topology

One of the most important initial steps in planning a Content Manager system is the planning and design of the system topology. There are various components in Content Manager, Library Server, Resource Manager, WebSphere Application Server, and database server.

Depend on your business needs, Content Manager system architecture can be build with a two-tier or three-tier configuration using different client options.

The two-tier configuration consists of Client for Windows or customized clients at the top tier, and the Library Server and Resource Managers at the bottom (see Figure 11-1).

![2-tier configuration diagram](image)

*Figure 11-1  Content Manager two-tier configuration*

The three-tier configuration consists of a browser at the top tier, eClient at the mid-tier, and the Library Server and Resource Managers at the bottom tier (see Figure 11-2).
Different Content Manager system configurations enable you to achieve different business requirements. For a first prototype system, you may install all components on a single machine. For an enterprise production system, you may choose to install each component on a separate machine, with more than one Resource Manager. See Figure 11-3 for a sample Content Manager system configuration on the AIX platform.
Many factors affect the final system configuration and topology for your Content Manager implementation. Among them, some important ones are the existing infrastructure, throughput requirements, performance requirements, service level agreements, and Business Volume Metrics. Your IBM representative has a Sizer tool you can use to make an initial rough sizing of the hardware requirements to support your workload. This helps you to finalize your system topology.

**Start Here CD**
The Content Manager Start Here CD is a useful resource to help your Content Manager planning and installation. The System Diagram section illustrates sample system topology diagrams for various types of configurations.

The Planning Assistant section interactively gathers your system requirements and generates a recommended plan that includes details about hardware and software requirements, product CDs needed, and installation and configuration steps.
11.4 Planning and designing data model

During the planning and designing phase, once a determination is made as to the type of content and the formats the system needs to handle, the next step is to plan and design an appropriate data model. A data model is the basic structure that is fundamental in storing and managing contents within a Content Manager system. A data model provides a template and can be thought of as being similar to a database schema.

Chapter 3, “Data modeling” on page 29 covers the details about data modeling. Because understanding data modeling is such an important aspect of planning and designing a Content Manager system, we briefly review the key concepts here in a slightly different perspective.

There are five key concepts in data modeling:

- Objects
- Items
- Item types
- Attributes
- Item relationships

An object refers to the actual electronic content stored. An item is a collection of information that identifies an object. For example, if a library has books and video tapes. You can digitize the books and the video tapes; the digitized content become objects. You can use a catalog to find these objects. An item in a catalog is not the object itself; but the item helps to uniquely and conveniently identify an object. In your Content Manager implementation, you need to collect information about objects in items. Items hold consistently formatted data that describes and identifies data objects.

An item type is a definition (or a template) of what data an item should collect as object information. In other worlds, an item is an instance of an item type. Attributes make up characteristics of an item type. For example, an item type called Book is defined by attributes such as Book Number, Title, Author, Published Date, and Subject. A specific redbook can be entered into the catalog by giving values for the various attributes that, either individually or together, identify a book.

Just as in any other data modeling, content manager gives the facility to conveniently relate items. You can link one item to another or have them reference each other. A book contains an author and therefore you can create a link between a book and its author. This saves you from having to enter duplicate information. Similarly, an attribute value in one item type can refer to an attribute value in another item type and can carry various delete rules (can be deleted, never deleted, cascaded) between them.
Planning for an optimal data model right from the beginning goes a long way in ensuring a successful Content Manager implementation. The steps that you have to go through to plan your data model include:

- Perform business analysis and identify all types of information to be stored in a Content Manager system. Examples include application forms, marketing brochures, and expense reports.
- For each item type, identify all the attributes that define the item. For example, and expense report have the following attributes: Employee ID, Amount, Date, and Description.
- Normalize your data model. For example, if employee ID is going to be used in many documents, Employee (Employee ID, First Name, Last Name, MI) should be created as a separate item and other item types can link or reference to it.

Depending on the item types and the data model you design, an existing Content Manager model can be used or a custom data model can be created.

**Important:** The System Administration Client in Content Manager V8.3 enables you to export your system administration data into and from XML. This new option applies for systems administration objects of Content Manager.

For more information on data model, refer to Chapter 3, “Data modeling” on page 29.

### 11.5 Planning and designing workflow

Along with data modeling, an important task of planning and designing a Content Manager system is designing the workflow for the system. Document Routing is the core business enablement feature of Content Manager Version 8. In previous versions of Content Manager, this feature was known simply as workflow.

Chapter 4, “Workflow” on page 75 covers the details about workflow. Because understanding it is such an important aspect, we briefly review the key concepts in this section in a slightly different perspective.

There are two key concepts in workflow:

- Process
- Work node
A *process* is a series of steps through which an item is routed. Map your department or organizational workflow to processes. You can create a variety of processes. Based on prior execution status and conditions, work in a process can branch off. The flow of work in a process can be serial, parallel, or branch off depending on the action of a user. A process moves documents or folders from one work node to another. A *work node* is a generic term referring to work baskets or collection points. It is the point in a process at which a user, an application, or an automatic action happens.

In general, while planning for a Document Routing process modeled after a business process, you need to keep the following steps in mind:

- Draw a basic flow of a process similar to a workflow chart. Individual steps in the process are the work nodes. Drawing a flow chart helps to visualize the flow before creating a process.
- Determine the nature of the flow: Continue or Escalate. This enables the process to branch off based on user or application actions. Remember that you can name these flows anything you want. For example, instead of Continue, you can change to Approve; instead of using Escalate, you can change it to Reject.
- Determine who is involved and has permission to work in a process. This helps you to define privilege sets or ACLs to be assigned to work nodes and processes.

**Note:** A process in Document Routing is created by using the System Administration Client or by using the APIs. With Content Manager V8.3, there is a graphical builder for creating Document Routing processes. Content Manager document routing is enhanced to include decision points, actions, action lists, parallel routing, and user exit support.

For more information on workflow, refer to Chapter 4, “Workflow” on page 75.

### 11.5.1 Hardware and software requirements

After you plan and design your Content Manager system, and before you install and configure the system, make sure you have the necessary hardware and software required for installing the components. There are separate requirements for installation of the various Content Manager components:

- Library Server
- Resource Manager server
- System Administration Client
- Client for windows
- eClient
You can install the components on various platforms: Windows, AIX, or Solaris.

For the minimum hardware and software requirements for installing Content Manager V8.3 and step-by-step instructions on installing it, refer to *IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332.

### 11.6 Capacity planning

In addition to meeting the minimum hardware requirements for a Content Manager system, you need to focus on planning for a scalable system. Your business drivers are performance, availability, and cost.

#### 11.6.1 Library Server capacity

Library Server is a main component of the Content Manager that controls access, provides transactions, and manages objects stored on one or more Resource Managers. Typically, most of your everyday user activity requests, such as update, retrieve, search, and delete, can be handled by the Library Server. Since the Library Server handles numerous peak hour requests for reads and writes, the machine where the Library Server is installed should be equipped with a powerful processor.

In addition, because the Library Server and its database must be installed on the same physical server, the machine should have enough space for the database, the program files, and any other prerequisite software. It is also prudent to plan on building database table indexes for custom tables created due to custom item types.

#### 11.6.2 Resource Manager capacity

Resource Manager is the repository for objects in the system. You should plan for adequate storage space. As a rough estimate, a baseline storage space should accommodate the number of total objects times the average object size.

The projected growth in the size and the number of objects should also be factored into the storage capacity calculations. A Resource Manager can be configured to enable LAN cache, so that the frequently accessed objects can be cached on a local Resource Manager, closer to the end user, regardless of which Resource Manager the object is originally stored at. (LAN cache is covered in 11.7.1, “LAN cache” on page 296.) Plan for a staging directory space on a Resource Manager accordingly.
11.7 Planning for performance

Performance and scalability of a system do not occur by themselves. There needs to be careful planning and consideration while laying out the system configuration. The Sizer tool and the Start Here CD give good starting points to build your configuration. Each Content Manager implementation is unique because of the workload, performance and scalability requirements. There are some best practices that can help you plan ahead of time for an optimally performing system:

- Understand clearly your project workload and performance objectives.
- Understand peak usage in terms of user operations and number of simultaneous transactions.
- Understand the frequently performed and significantly resource intensive operations performed by typical users.
- Perform simulated load and stress tests to get satisfactory TPS and response time metrics.
- Fine-tune the system during the initial test period or rollout, focusing on one bottleneck or improvement area at a time.
- Plan on continuous performance monitoring once the system is in production:
  - Perform periodic database, file system, and software configuration tune-ups as described in respective sub-systems.
  - Use monitoring tools to monitor key performance indicators such as CPU, memory, and disk utilization.
  - Proactively monitor metrics and compare against project growth of the system to avoid issues later.

Adequate system performance is both an exit criteria as well as a project constraint. Achieving a highly performing system involves trade-offs. It is important to understand the trade-offs and choose options that are more relevant to business requirements. Next, we discuss some of these options and how they impact performance.

**Web clients versus desktop clients**

There are basically two options to implement your Content Manager client that a user would use to access the system: Client for Windows and eClient. Client for Windows runs on a standard Windows desktop; whereas the eClient can be run from any supported Web browser. The desktop client is typically faster than the Web clients, while the Web clients are typically easier to deploy and maintain.
With the eClient choice, there are further options to choose. You can either connect directly or connect through a mid-tier conversion server. Direct connection is faster but may require browser plug-ins or viewer applets to directly view documents.

**IBM default components versus custom components**

Depending on the business requirements, you may choose to build your own applications or components. In this situation, it is important to know that the IBM default components that come with the package are already tuned for performance and scalability in a generic way. There is always the possibility that the custom applications give you more control and can be tuned to match your exact requirements.

A typical example of a custom application is building a client for the Content Manager. The custom client can be built using the APIs and exit routines. For custom clients that are built using Java or C++ APIs, both the document data model and the custom model options are available.

**Hardware trade-offs**

Your business volume metrics and performance requirements determine the final system configuration you choose. Separate machines for the Library Server and the Resource Manager result in higher scalability and performance. Multiple Resource Managers provide for higher network bandwidth and hence optimize object transfer. Distributed Resource Managers that are geographically close to the end users provide for greater performance and availability.

**Other regular Content Manager features**

Certain options, if you choose carefully, can help you to avoid future surprises. Versioning gives the user the ability to work with prior versions; but versioning increases the Library Server database size and also degrades performance while accessing prior versions (comparing to access the most current version). Maintaining appropriate database indexes improves searches and the Library Server performance; but maintaining a large index collection may increase the size of the Library Server database.

There are two other key Content Manager elements that affect performance, scalability, failover, and availability. These are the LAN Cache and Replication features. Next, we provide a brief overview of these two features.

### 11.7.1 LAN cache

Your Content Manager system configuration can involve more than one Resource Manager to manage digital content. A Resource Manager can be located geographically close to the user and serve as local server. Whenever an
object is not found on the local server, it is retrieved from the appropriate remote Resource Manager and stored in the staging directory of the local Resource Manager server. This is an optional feature and is known as *LAN cache*.

Enabling this option helps to minimize accessing remote Resource Managers for frequently accessed objects. This improves performance and helps avoid network bottlenecks. Enabling LAN cache mandates more staging directory management tasks that need to be performed by a system administrator. Some of these tasks include:

- Defining the size of the staging directory.
- Defining the maximum size of a cached object.
- Setting automatic purge of least frequently used objects in the staging directory.
- Setting up subdirectories to hold cached objects. This speeds up searching by targeting at a subdirectory level rather than looking at individual objects.

### 11.7.2 Replication

Depending on the workload requirements, your Content Manager implementation may have more than one Resource Manager server. To provide for failover and reliability, objects can be replicated from one Resource Manager to another. This is vital when the original Resource Manager is unavailable for some reason and a backup server is required to support regular operations.

Data is stored in a Content Manager system in terms of items and objects. Items are stored in entities called collections. A *collection* is a grouping of items that have the same storage groups and migration policies. The items in a collection are stored on the same storage system and are migrated based on set policies. Collections are created on each of the Resource Managers.

Replication involves moving collections from a primary Resource Manager to one or more backup Resource Managers. For example, a collection on a primary server can be enabled for replication to a backup server. A collection on the backup server can be enabled to receive replication from the primary server. It is recommended that collections be defined to distinguish those that are replicated parts, versus those collections that are primary copies. Replication may be configured to occur during off-peak hours to reduce load on the servers.

In order for replication to work, the Library Server, as well as the primary and secondary (backup) Resource Manager servers, need to be defined and have visibility to each other. Content Manager provides a service that monitors the failover of the Resource Managers.
11.8 Planning and designing text search

The text search feature is an option in Content Manager V8.3. This feature lets users search for documents in the system using words or phrases. The text search function automatically indexes, searches, and retrieves documents stored in the content manager system. A prerequisite for running text search function is to install the Net Search Extender as along with the DB2 database software while installing the Library Server.

For information on text search, refer to Chapter 5, “Text indexing and searching” on page 117.

If your Content Manager runs on z/OS, the text search feature is not available.

11.9 Planning and designing security

Security plays a key role in any organization’s infrastructure. After you install and migrate content to a Content Manager system, you do not want unauthorized users to access the system, nor do you want users to perform operations that they are not supposed to do. Authentication and authorization form the core steps in providing security to your Content Manager system. Authentication is the process of verifying that the user really is who they say they are. This is accomplished by checking their user login and password against an authentication repository.

You can plan to use an existing IBM LDAP Directory where your system users have already been defined. The Windows Active Directory can also be used as an authentication source. A third possibility is using the Lotus Notes Address Book to authenticate users.

Once users are authenticated, they are given access to the Content Manager system. The system security then authorizes what the users can do in the system with privilege sets and ACLs. A privilege is a generic high level system permission allowing a user to create, retrieve, and delete objects. An ACL sets more restrictive access on an object. For example, you grant users the right to retrieve certain objects, but you do not allow them to delete any of the objects.

From the initial business analysis, you should know who, from which business department or area, are the owners and users of what content. This analysis groups users and enables the assignment of privilege sets and ACLs based on user groups. This process also results in easier maintenance and management of security on the system.

For more information on security, refer to Chapter 8, “Security” on page 187.
11.10 Options checklist

Implementing a Content Manager system involves careful planning, analysis, and design. You need to choose available options depending on the business needs and your unique project requirements. Each option has its own implications. Here is a brief checklist to help you organize your planning:

- Is sufficient information available to answer questions highlighted in 11.2, “Analyze business operations and requirements” on page 286? Well scoped business requirements help in successful implementation.

- Make sure your architecture and system topology are finalized and consistent with the existing infrastructure. You may have to order more hardware or software, depending on the configuration.

- Which platform to use: Windows, AIX, Solaris, or z/OS? There are several implications, depending on which operating system supports your Content Manager implementation. For example, in z/OS environment, there is no text search support yet. Again, you want to choose a platform that is consistent with the existing infrastructure, or the one that you are moving forward to.

- What database to use: Oracle or DB2? Installation, fixpacks, and software requirements vary depending on which database you choose.

- Which Content Manager Client to use? As noted in the performance section, there are differences in performance characteristics, depending on whether you decide to use Client for Windows, eClient, or a custom client as your Content Management application client. Also, there are functionality differences. For example, while building the data model, references cannot be seen with Client for Windows but with the eClient.

- Which Content Manager versions to use? If you are migrating from an earlier version of Content Manager, it is possible to run the old and the new versions in parallel, thus saving hardware resources. For more details, please consult Part 4, “Content Manager migration” on page 345.

11.11 Summary

This chapter has introduced you to the planning and designing of a Content Manager system. Topics included planning from a business perspective, planning and designing the data model and workflows, planning and designing the system topology, hardware and software prerequisites, and performance requirements. With this knowledge as background, the next chapter leads you into the installation and configuration of a Content Manager system.
Deployment

In this chapter, we discuss typical activities involved in deploying a Content Manager solution in a production environment. Taking a project-oriented approach, we cover how system verification, testing, software configuration management, and code updates should be handled during the course of a Content Manager solution deployment.
12.1 System verification

For installation of Content Manager V8.3, refer to the product manual:

*IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332

After having gone through the steps required to install and configure Content Manager components, you should have a successfully installed the system. An important task of deployment is making sure your system is up and running as expected.

For your convenience, we briefly review some of the high level steps required to verify a successful installation. You need to verify the following components to ensure that your system is successfully installed:

- **Library Server**
  - Library Server database
  - Library Server monitor

- **Resource Manager**
  - Resource Manager database
  - Resource Manager application
  - Resource Manager services

- **eClient application**

- **Client for Windows application**

12.1.1 Verification utilities

You can verify that your system is successfully installed by using the following two utilities:

- **First Steps**
- **Installation validation utility**

*Using First Steps*

If you have installed all the Content Manager components on a single machine, use the First Steps program (Start → Programs → IBM DB2 Content Manager Enterprise Edition → First Steps) to load sample data. In the case where the components are installed on multiple machines, use the Server Configuration Utility to connect to the appropriate database and load the sample data.
Once you have loaded sample data, you can use the System Administration Client to log on. A successful logon indicates a successful connection between your System Administration Client and the Library Server; it also indicates that the Library Server database is setup properly.

Navigate to **DB2 Content Manager → icmnlsdb → Data Modeling → Item Types** and confirm that some XYZ item types have been created.

Open the Windows Client or the eClient, search for a document, and open it. If you can view the document, then both the Library Server and the Resource Manager are up and running.

You can also see if the sample data is loaded properly in the Resource Manager data directory (LBOSDATA). This is default location where all Content Manager data is stored. For example, in a Windows machine, if the mount point is C:\, then all DB2 Content Manager data is stored in
C:\lbosdata\collection_ID\obj_path\obj_itemid

**Using installation validation utility**
You can run the installation validation utility to check that the installation, configuration and deployment have been successful. In the validation utility, you can check:
- Content Manager
- Information Integrator for Content
- Content Manager eClient

Start the validation utility by selecting **Start → Programs → IBM DB2 Content Manager Enterprise Edition → Installation Validation Utility**.

You need passwords for the Library Server administrator ID (icmadmin) and Resource Manager ID (rmadmin). The utility should return “successful” on each validated component.

### 12.1.2 Verify individual components

You can manually verify that the individual Content Manager components are installed successfully, in addition to using the verification utilities as discussed in the foregoing section.

**Verify Library Server installation**
Verify that the Library Server database is installed successfully. Make sure you are able to connect to the database using the appropriate user IDs. For example, to check connectivity for user icmadmin from a DB2 command window:

```
db2 connect to icmnlsdb user icmadmin using password
```
Check the installation log file to see if any errors logged during installation.

**Note:** In Content Manager Version 8.3, all installation log entries are put in one log file:

```
%IBMCMROOT%/log/cminstall.log
```

If anything goes wrong, more detailed information will be put in the component specific log directory under:

```
%IBMCMROOT%/log/
```

Verify that the tables are created in the Library Server database. For DB2, these tables should have names starting with “FA” and “ICM”. For Oracle, table names always start with “ICM”.

**Note:** In Content Manager Version 8.3, the C++ compiler dependency has been removed. All item creation, reading, updating, and deletion are performed with dynamic SQLs. You do not need to check for the creation of DLLs anymore.

The Library Server monitor program is installed during the Library Server installation. This program detects the availability of Resource Managers defined to the Library Server. It is installed as a service (icmplasp) on Windows and as a started process on AIX and Solaris.

**Verify Resource Manager deployment**

Verify Resource Manager database connectivity using the appropriate user ID. For example, check connectivity for user rmadmin from a DB2 command window:

```
db2 connect to icmnlsdb user rmadmin using password
```

Check to see if the tables have been created. Make sure that there are no critical errors logged in the cminstall.log during the installation process.

By using the DB2 command, DB2 Application Listing, check to see if three RMDB java.exe processes are running:

```
db2 list application
```

Resource Manager is a WebSphere Application Server application. Open a command prompt and change directory to `%WASROOT%/bin`. Issue the following command to verify that the Resource Manager application is running:

```
serverstatus icmrm
```
To further verify that the Resource Manager Web application is installed properly, open a Web browser and type the following address:

http://<hostname>/icmm/snoop

**Verify eClient installation**

If your solution includes an eClient installation, you can verify if it is set up successfully. Resource Manager is a WebSphere Application Server application. Open a command prompt and change directory to %WASROOT%/bin. Issue the following command to verify that the Resource Manager application is running:

```
serverstatus eClient_Server
```

Start eClient application from your browser, and retrieve some sample data loaded from First Steps:

```
http://localhost/eclient/IDMInit
```

**Verify Client for Windows installation**

Go to **Start** → **Programs** → **IBM Content Manager V8** → **Client for Windows**. Log on to the Library Server. In the Welcome panel, click **Search** button. Choose the Auto Photo item type which is a sample item type loaded in V8. Use a wildcard (*) in the Adjuster Last Name search field and click **OK**. From the results, double-click on the item to view the image.

### 12.1.3 Post-installation changes

There are usually some post-installation changes necessary due to various reasons. They include insufficient information during installation, unexpected errors, or even optional changes. In this section we discuss, at a high level, some typical post-installation changes that may occur.

In case your Library Server or Resource Manager databases are not created properly, there are utility programs available to create or replace Library Server and Resource Manager databases. For Windows:

- For creating or replacing a Library Server database utility, use:

  ```
  C:\Program Files\IBM\db2cmv8\config\cmcfgls
  ```

- For creating or replacing a Resource Manager database utility, use:

  ```
  C:\Program Files\IBM\db2cmv8\config\cmcfgrmdb
  ```

Similarly, if the Resource Manager WebSphere Application Server application is not installed automatically, you can run the following utility:

- For creating and configuring Resource Manager application and deploying into WebSphere V5 utility, use:

  ```
  C:\Program Files\IBM\db2cmv8\config\icmrmcfg5
  ```
Another alternative is to use the installation program and then select the component you would like to re-create (the Library Server database, Resource Manager database, or Resource Manager application).

To manually start the Library Server monitoring program, run the `icmserv.exe` command on Windows. On AIX and Solaris, run the script under `/etc/rc`.`cmprmproc`.

IBM License Use Management (LUM) is a program to monitor software licenses dynamically. LUM can be installed for free by downloading it from the following Web site:

http://www.ibm.com/software/lum

You should configure LUM for Content Manager by using the Configuration Tool: Click Start → Programs → License Use Runtime → Configuration Tool.

If you need to uninstall any of the Content Manager components for any reason, follow the steps for your platform. For example, on Windows, you can use the Add/Remove Programs in the Control Panel to remove selected components for Content Manager V8.3.

**12.2 Deploying custom applications**

You may need to make many customizations to the standard Content Manager functionality. This may involve creating advanced data model, building a custom client for Content Manager, or integrating Content Manager with an external application. Make sure to move, register, or activate all of that custom functionality on the appropriate component servers.

**Note:** If you did not create or convert your custom application, you need to analyze your existing Content Manager system and determine how it fits with the new version. We strongly recommend reviewing Part 2, “Understanding the product” on page 27 to learn detailed information on data model, workflow, text indexing and searching, application development overview, query language, security, and optionally the TSM overview. Also refer to Chapter 17, “Application migration” on page 451 to help migrate your application.
12.3 Testing

Testing is a key activity which assures that the system conforms to all business expectations. There are many business requirements that are part of a typical project charter, including functionality, integration, performance, and failover recovery. It is not uncommon that organizations may run many tests before deploying a system to production. To support the development and testing of a Content Manager solution, companies should have, at a minimum, two separate environments, as shown in Figure 12-1.

![Figure 12-1 Environment setup](image)

Configuration, solution construction, and integration activities happen typically on a unit and/or development environment. This environment provides for the sandbox that the developers use to install, configure, and develop solutions with Content Manager. All Content Manager components can go to a single machine or on different machines in a development environment. This depends on the availability of sufficient hardware, cost, and other infrastructure constraints. The development environment is where solutions are taken from the design phase into construction and development phase of the solution. Unit testing is performed to verify that solutions indeed satisfy the functional requirements.

The solution needs to be tested for performance, scalability, and failover. Since the unit environment is always in a state of flux, usually a different environment with separate machines should be used. This environment is referred to variously as a system, performance, or simulated production environment. This separate environment needs to be as close to production as possible. This is where you conduct performance, stress, load, and failover tests. It is also sometimes called a simulated production environment because it represents the actual production system and is used to simulate and fix any defects.
Some organizations do have more environments, depending on how elaborate their infrastructure is and how extensive their IT needs. Many of these are put together on an ad-hoc basis and disbanded after their purpose has been served. Plan ahead of time to ensure that you always have a test environment to test your fixes and upgrades.

12.4 Software configuration management

As we discussed in the previous section, you need multiple Content Manager environments to build and support a solution. How do you make sure that the right version propagates from environment to environment and ends up as the right solution in production? This calls for a very efficient software configuration strategy for your Content Manager implementation and migration project. You need to use a standard off-the-shelf configuration or source code management tool such as PVCS, ClearCase®, or SourceSafe. Make sure your SCM structure is easier for the build and code deployment process.

All your custom code should go as part of an SMS issuance or package. Having a package that automatically installs custom code greatly helps in cases where you have to reload development servers. Furthermore, in cases of server failures in production, having a readily available Content Manager package is extremely useful after restoring the server and bringing up the Content Manager solution.

12.5 Production rollout

Rolling out a system in production requires careful planning and execution. It is better to phase the rollout in different stages so that you can bring in a relatively small number of users on the system initially. This enables you to check out if the infrastructure is fine, response times are acceptable, and let the application support group build accurate support solutions to handle help desk calls. If you are employing proactive system management techniques to monitor for hardware or software failures, do a trial run.

Having a readily tested infrastructure before actually rolling out to users helps to identify bottlenecks. For example, you can test for failure of a Resource Manager and check how the Library Server responds to it. Load balancing and failover tests should be performed before turning on the production solution.

Refer to Chapter 20, “Performance tuning” on page 543 for performance related issues.
12.6 Updates

Once your Content Manager system is in production, there can be many types of updates to the system. Functionality enhancements, extensions, integration, and maintenance of the solution becomes critical. Proper configuration management enables you to test and move upgrades in a systematic manner.

Make sure you apply the most recent fixpacks for Content Manager on a regular interval. The fixpacks can be downloaded from:

http://www.ibm.com/software/data/cm
Case study

In this chapter, we discuss the implementation of a Content Manager solution for a fictional company, ACME Marketing. We go through all the required steps from designing the architecture and data model, through implementing the Document Routing and migration policies. By using the case study scenario, we hope to give you a better understanding of how to successfully implement a Content Manager solution, building on the concepts covered in the previous chapters.
13.1 Introduction

The following case study discusses ACME Marketing, a fictional marketing company. ACME Marketing provides marketing services to other companies, primarily in the form of advertisement production. Included in these services is the production of various marketing campaigns that include advertisements for:

- **Television**: This includes advertisements that appear between and during television shows.
- **Radio**: This includes advertisements that are played between songs and other programs on radio.
- **Movie**: This includes advertisements that are shown at the beginning of movie screenings, prior to or after the movie trailers.
- **Print media**: This includes advertisements that appear in newspapers and magazines.

ACME Marketing is a large multi-national organization with major offices located in New York, London, and Sydney. Although each office generally focuses on developing marketing campaigns for its own region, the offices often work together to produce international campaigns for their multi-national customers. Existing marketing materials for customers in one country may also be reused or customized for the same customer in another country. For example, an advertisement may use baseball to convey a message towards the sports fans in America; it may use cricket for the fans in England or Australia. ACME Marketing uses a large pool of shared, generic resources, such as landscape photographs, to form part of a variety of advertising campaigns.

13.2 Business problem

ACME Marketing currently have no managed system to store and maintain their marketing materials. Advertisements are generally stored on some arbitrarily shared file systems or even locally on marketing executives’ machines. Sometimes, the materials are transferred via e-mail among the employees to enable collaboration. Likewise, when quality assurance and legal reviews are required, the marketing materials are transmitted via e-mail to the appropriate reviewers, leaving e-mail transcripts and a manual, paper-based system as the only way to track the approval history of any given advertisement.

In the beginning, when the company has only one New York branch, this system enabled the employees to perform their business with reasonable effectiveness. As the company grows, with new offices in Sydney and London, the creation and the tracking of the marketing materials have become an administrative nightmare.
These are some of the problems ACME Marketing has experienced:

- The heavy reliance on transferring marketing materials via e-mail has led to overloading of the mail servers. Some of these mail files sizes are huge; the sheer size of the large files has made it virtually impossible to transmit them via e-mail.

- It is difficult to efficiently track down the latest version of the marketing materials. With multiple copies existing in multiple e-mails, on various shared file systems, or local drives, these materials, particularly their latest versions, are often very difficult to locate.

- E-mail transcripts have proven inadequate in tracking quality assurance and legal approval of the advertisements. There is virtually no traceability of these advertisements in terms of workflow, modification history, or approval history.

- No security is enforced at the file-system level.

It has become apparent, that unless something is done to address these issues, the company’s effectiveness in producing advertisements could be severely inhibited. The decision has been made to implement a Content Manager system that will be responsible for:

- The storage and retrieval of all advertisements
- The storage and retrieval of media resources (generic resources that are used to produce new advertisements)
- The quality assurance and legal approval of advertisements
- The enforcement of content security
- The tracking of document modification and approval history

13.2.1 Requirements

Using a series of questions, as discussed in 11.2, “Analyze business operations and requirements” on page 286, this section breaks down the very high level requirements outlined above into a more detailed set of requirements, both functional and non-functional.

Note: Currently, most employees in the new offices (Sydney and London) are focused on marketing, with very few IT employees assisting them at the local help desks. The New York office, the main office, has a large IT team. It is this IT team that will administer the new system for all the offices.
**What types of content need to be managed?**
The system needs to manage the storage and retrieval of two types of content, advertisements and media resources, for:

- Television
- Radio
- Movie
- Print media

**What are the various electronic formats used and planned?**
The advertisements and media resources may have these electronic formats:

- **PDF:** This is the only file format allowed for print media. Any final version of the advertisements in a magazine promotion must be in the PDF format.
- **MP3:** This is the format used for audio files. They are used for radio advertisements.
- **MPEG:** This is the format used for all video files. They are used for television and movie advertisements.
- **GIF:** This format is one of the accepted formats for image files. It makes up part of the media resources.
- **JPEG:** This is another acceptable format for image files. It makes up part of the media resources.
- **Text:** Some files may be stored in native text format. These may be used to store the transcripts of audio or video files.

**What is the expiration and retention policy of the content?**
ACME Marketing would like to store all advertisements and media resources indefinitely. The advertisements that are created within the last year need to be readily accessible. Similarly, the media resources that are created within the last 5 years need to be readily accessible. Delay in retrieving other content is acceptable.

**What are the estimated sizes of the active electronic content?**
The file sizes of the content vary dramatically depending on the type of content. For example, video files may be as large as 100 MB, and text files can be as small as 1 Kobe.

**What is the projected growth rate over the next three years?**
Based on the current rate, the New York office adds about 300 MB of advertisement to the system per day. Other offices usually add about 100 MB each to the system. This totals to approximately 500 MB per day for advertisements for all offices. In addition, the three offices collectively produce approximately 50 MB of new media resources to the system every day.
Assuming that the business will expand 20% every year, and we have 300 productive days in a year, this translates to approximately 165 KB for the first year (550 x 300 days), and 600 KB (165 + 1.2 X 165 + 1.2 x 1.2 x 165) for the next 3 years.

**Who will manage and use the content?**

All company employees should have at least read-only access to the system. In addition, the following groups of users need various levels of access:

- **Content creators:** They are owners of the content. They use and manage the content, and have all the rights prior to the content approval by the legal reviewers.
- **Quality assurance reviewers:** They are users of the content. They are not allowed to modify or delete the content.
- **Legal reviewers:** They are users of the content. They are not allowed to modify or delete the content.
- **System administrators:** They do not use the content, but they can manage the content. They have all the rights related to the content.

**What are the workflow requirements?**

To begin with, only one standard workflow is required. This is to support the creation and approval of the advertisements. Users are not allowed to delete content at any stage. If deletion is required, the system administrators can perform this action upon request and can only do so prior to the content approval by the legal reviewers. Once the legal reviewer approves the content, the workflow is completed and no deletion is allowed.

The creation of media resources does not require a workflow, with the only limitation being who can create them.

Another temporary workflow is required to allow the import of the existing advertisements. These advertisements do not need to go through any review stages; but they need to be specially marked as being created prior to the new Content Manager system. Note that this temporary workflow is not covered in this case study.

**What are the migration requirements?**

Since there is no existing Content Manager system or anything other systems in place, no migration effort is required. The very random nature of the current location of advertisements and media resources means that the current owners of the existing content are expected to import their content into the new system manually. This requires the creation of the temporary workflow as mentioned earlier. We choose not to cover this in the case study.
**How many users are expected in the final rollout of the system?**
It is estimated that about 1,000 users are required to access the system. At any
given time, however, there are no more than 100 concurrent users.

**What are acceptable response time and throughput requirements?**
Due to the distributed nature of the company, the system is expected to handle
about 100 concurrent users, 8 hours a day (during the standard New York
business hours) and 50 concurrent users for the other 16 hours in a day, due to
the smaller offices in Sydney and London.

While most users are active with the intention of viewing content, the system is
expected to easily accommodate the submission of 30 pieces of content an hour,
during the peak usage. ACME Marketing also requires:
- Search response time within 10 seconds
- Page launches (containing meta data, not content) within 6 seconds

Due to the significant variation in the sizes of the content stored in the system,
the response time associated with the content retrieval differs dramatically. It is
expected, however, that when a content is stored locally (that is, in the same
office as the user), the retrieval of a 10 MB file should take no more than 20
seconds.

**Note:** Obviously, there is a large dependency on network speed for these
performance requirements. For the purpose of this case study, we assume
that there is a high-performance network and that network traffic does not
need to be considered.

**What is the existing infrastructure?**
Currently, ACME is using Windows as clients and servers, with some shared file
systems, and Lotus Notes e-mail servers. ACME Marketing has decided that AIX
is the strategic platform that the company wants to move to, with DB2 as its core
database management system.

**Will a custom data model be required?**
A custom data model will definitely be required. This is discussed in great detail,
in 13.3.2, “Data model” on page 322.
What are the versioning requirements?

It is not expected that there will be a heavy requirement for versioning of content. Generally, only the final versions of advertisements are stored and there is rarely a need to modify them; however, ACME Marketing wants the ability to store up to three versions of an advertisement at any given time. There is no requirement to versioning the media resources.

Are there any other considerations?

Since the system will be a pivotal component of the company’s day-to-day business, it is important for ACME Marketing to have some protection against system down time. It is acceptable, however for the system to perform at a reduced capacity during this temporary problem.

Although the system is to be used for the advertisements and media resources, ACME Marketing would like the system to be designed such that in the future, the system can be extended to other parts of the business if required. Similarly, they would like the system to be scalable so that, as the business grows, it can be extended to handle the extra workload.

Finally, ACME Marketing would like the system to be flexible so that potential new offices can have access to the system, with users experiencing similar levels of performance and functionality as the users in the existing offices.

13.3 Designing the solution

Now that we have a better understanding of the requirements, we can design a solution to meet those requirements. The following sections address the various components of the system that we need to consider.

Note: This is an iterative process, as many components are interdependent. As a result, you may create an initial architecture and then return to make subtle changes once your plans have been finalized.
13.3.1 Architecture

When designing the architecture for the Content Manager solution, there are many important factors to consider:

- The distributed nature of the users:
  - Need to collaborate on advertisements despite this distribution.
  - Performance levels need to be adequate in all offices.
- Limited failover requirements.
- Flexibility to grow in terms of users and offices
- Choice of strategic platform and database (AIX and DB2)
- Potential hardware budget restrictions
- Location of IT administrative team in New York office only

With no restrictions in terms of budget and skill resources, the architecture would contain a large number of servers with:

- One central Library Server
- Multiple Resource Managers (one at each office)
- Multiple backup Resource Managers (one for each office's Resource Manager)
- Multiple TSM servers (one at each location)

ACME Marketing has a limited budget and would like to keep as many administrative tasks at the New York office as possible. This architecture, therefore, needs to consolidate on server numbers in an attempt to find a logical compromise between server numbers, system performance and system availability.

The following figures demonstrate the architecture to be implemented by ACME Marketing.
Figure 13-1 shows all the required servers, highlighting the replication of all three Resource Managers to a central backup Resource Manager.

**Note:** All components are explained in greater detail later in this case study.
Figure 13-2 shows the same architectural components, but highlights how objects are *migrated* to a central TSM server.

**Library Server**

The Library Server will be located in the New York office, and it will be used by all three offices. It will be called ICMNLSDB.
**Resource Managers**
There will be three primary Resource Managers for the ACME Marketing Content Manager system, one at each office (SYDRMDB1, NYCRMDB1, LONRMDB1). LAN cache will be enabled on each Resource Manager and the TSM client APIs will be installed to allow for migration from each Resource Manager to a central TSM server.

Users from each office will be setup to have their local Resource Manager as their default. This means Sydney users will have SYDRMDB1 as their default Resource Manager, New York users will have NYCRMDB1 as their default, and London users will have LONRMDB1 as their default Resource Manager. With LAN cache enabled on each server, objects stored on remote Resource Managers will be cached to a user's local Resource Manager after it is retrieved for the first time. Note, a custom application can be written to “pre-fetch” these objects so that they are automatically cached, and all users consequently will experience better retrieval performance.

A backup Resource Manager will be located in New York. It will serve as a failover for each of the other three Resource Managers as all objects will be replicated to this server. If one of the primary Resource Managers goes down, users will automatically start using the replica Resource Manager. For Sydney and London users, this may mean some performance degradation; but since this will only be temporary while the local Resource Manager is brought back up, this solution should be acceptable. With less budgetary constraints, it is possible to set up a backup Resource Manager at each location, removing the chance of this temporary performance issue.

**Important:** Although this backup Resource Manager will be in New York, it should be in a different location to the primary server. In this way, if some disaster such as fire destroys the primary server, the backup should still be available.

The decision to use three collection on each primary Resource Manager is a decision made primarily for replication and migration purposes. This decision will be explained later, in 13.3.4, “Migration and document life-cycle” on page 331.

**Tivoli Storage Manager**
One central TSM server will be used by ACME Marketing for migration from all three primary Resource Managers. Similar to the decision to have one backup Resource Manager, this one central TSM decision is made because:

- ACME Marketing likes to consolidate on the number of servers initially used.
The Sydney and London offices have no IT administrative team; this means that ACME Marketing needs to keep as many “administrative tasks” at New York, one central location, as possible.

With no budgetary constraints, ACME Marketing can potentially have a TSM server at each location.

13.3.2 Data model

The data modeling goals for ACME Marketing are to reduce data redundancy and to enable a relatively simple form of classification and categorization of each advertisement and media resource.

For each media resource, they would like to know the following information:

- Media type (for example, “Print”, “Media”)
  - The user should be able to select from a list of valid media types.
- Description
- Current status (for example, “Open”, “Closed”)
- Creator Name
- Creation Date

For each advertisement, they have the following data requirements:

- Campaign information that includes:
  - Campaign name
  - Campaign description
  - Campaign start date
  - Campaign end date
  - Customer information that includes:
    - Customer name
    - Customer description
    - Customer industry type
- Advertisement name
- Advertisement description
- Advertisement transcript (if applicable)
- Advertisement start date
- Advertisement end date
- Advertisement status (for example, “Awaiting Approval”, “Approved”): This should be restricted to one of a list of valid status.
- Media type (for example, “Print”, “Media”):
  - The user should be able to select from a list of valid media types.
- A list of any media resources that are used for this advertisement
A list of contributors to this advertisement, including:
- Contributor's name
- Contributor's main area of expertise
- Description of contributor's actions
- Contribution type (for example, “Producer”, “Author”)
  This should come from a list of valid contribution types.

A list of products advertised, including:
- Product name
- Product description
- Brand information:
  - Brand name
  - Brand description
  - Customer information

The important thing to note here is that many entities, such as customers, brands, products, and contributors, will be used in multiple places and potentially multiple times. To reduce the amount of times the same data is entered (thus reducing the data redundancy), we have applied normalization techniques, much the same as we would if we were doing standard database design.

The data model that ACME Marketing has finally decided upon is shown in Figure 13-3.
Keyword

The item type *Keyword* consists of lists of keywords that contain valid attribute values for other item types. Each keyword list shares a GroupID and has a unique KeywordID. For example, the list of acceptable industry types, such as Information Technology and Retail, shares a common GroupID, IND_TYPE.

Each value has a unique KeywordID, such as IT_001 and IT_002. In this way, when a user needs to classify a customer, the user can choose from a list of valid industry types (all keywords with GroupID equal to IND_TYPE) for the customer. For presentation purposes, the keyword description is displayed to the user; but the KeywordID is stored with the customer item. Consequently, any updates to the keyword description is immediately reflected in the customer item.
Table 13-1 lists the attributes of the Keyword item type.

**Table 13-1  Keyword attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>The identifier that relates a logical group of keywords (example, Status).</td>
</tr>
<tr>
<td>KeywordID</td>
<td>The unique identifier of a keyword.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of a keyword for display purposes.</td>
</tr>
</tbody>
</table>

In this case study, Table 13-2 lists the pre-populated keyword groups to be used in the system.

**Table 13-2  Keyword groups**

<table>
<thead>
<tr>
<th>Group ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND_TYPE</td>
<td>All keywords specifying valid industry types (example, Information Technology and Retail).</td>
</tr>
<tr>
<td>STATUS</td>
<td>All keywords specifying valid status (example, Approved and Rejected).</td>
</tr>
<tr>
<td>MEDIA_TYPE</td>
<td>All keywords specifying valid media types (example, Newspaper and Television).</td>
</tr>
<tr>
<td>CONTR_TYPE</td>
<td>All keywords specifying valid contribution types (example, Producer and Composer).</td>
</tr>
</tbody>
</table>

**Note:** While implementing a keyword item type is not absolutely required, it is an effective way of mandating what keywords can be used by users and an easy way of managing changes to the keyword descriptions.

**Customer**

The item type *Customer* is classified as an item. This means that no object can be stored with it, similar to that of the Keyword item type.

Table 13-3 lists the attributes of the Customer item type.

**Table 13-3  Customer attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerID</td>
<td>The unique identifier of a customer.</td>
</tr>
<tr>
<td>IndustryTypeID</td>
<td>The identifier of the industry type of the customer. This is a foreign key, linked to the Keyword item type.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
<td>The customer's name.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the customer.</td>
</tr>
</tbody>
</table>

**Brand**

The item type *Brand* is classified as an item.

Table 13-4 lists the attributes of the Brand item type.

**Table 13-4 Brand attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BrandID</td>
<td>The unique identifier of a brand.</td>
</tr>
<tr>
<td>CustomerID</td>
<td>The unique identifier of the customer that owns this brand. This is a foreign key, linked to the Customer item type.</td>
</tr>
<tr>
<td>Name</td>
<td>The brand name.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the brand.</td>
</tr>
</tbody>
</table>

**Product**

The item type *Product* is classified as an item.

Table 13-5 lists the attributes of the Product item type.

**Table 13-5 Product attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductID</td>
<td>The unique identifier of a product.</td>
</tr>
<tr>
<td>BrandID</td>
<td>The identifier of the brand that this product is part of. This is a foreign key, linked to the Brand item type.</td>
</tr>
<tr>
<td>Name</td>
<td>The product name.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the product.</td>
</tr>
</tbody>
</table>
**Employee**

The item type *Employee* is classified as an item.

Table 13-6 lists the attributes of the Employee item type.

*Table 13-6  Employee attributes*

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmployeeID</td>
<td>The unique identifier of an employee. This is a foreign key to an external DB2 database which currently stores all employees details.</td>
</tr>
<tr>
<td>UserID</td>
<td>The Content Manager user ID of this employee.</td>
</tr>
<tr>
<td>FirstName</td>
<td>The first name of the employee.</td>
</tr>
<tr>
<td>LastName</td>
<td>The last name of the employee.</td>
</tr>
<tr>
<td>ExpertiseArea</td>
<td>A description of the employee's area of expertise.</td>
</tr>
</tbody>
</table>

The employee item should be created after registering the user in Content Manager. The UserID attribute gives us a way of mapping system user information back to ACME Marketing's more detailed Employee item type. For example, Content Manager automatically stores the user ID of the creator of any item. With this information, we can find the associated Employee item to get additional information about that person. This approach gives ACME Marketing the flexibility to add more attributes to the employee item type in the future.

**Note:** This is not the only way to implement this requirement. For example, if ACME Marketing is going to integrate with LDAP, we can refer to information stored in the LDAP repository using some custom code, rather than requiring an Employee item type. This is just the simplest and easiest way to satisfy ACME Marketing's requirements.

**MediaResource**

The item type *MediaResource* is classified as a resource item. This means that we can store an object against the meta data that we define. It uses the DKLobICM media object class so that it has the flexibility to store any type of object.

Table 13-7 lists the attributes of the MediaResource item type.

*Table 13-7  MediaResource attributes*

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaResrcID</td>
<td>The unique identifier of a media resource.</td>
</tr>
</tbody>
</table>
The item type *Campaign* is classified as an item. It is implemented as a folder that contains all the advertisements that are used as part of any particular campaign. For example, a campaign item with the name of Campaign X is created, using the semantic type of folder. Any advertisement that is created for this campaign is automatically put into this folder, using auto-linking. This is described further in “Advertisement” on page 328.

Table 13-8 lists the attributes of the Campaign item type.

### Table 13-8  Campaign attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CampaignID</td>
<td>The unique identifier of a campaign.</td>
</tr>
<tr>
<td>CustomerID</td>
<td>The identifier of the customer for the campaign. This is a foreign key, linked to the Customer item type.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the campaign.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the campaign.</td>
</tr>
<tr>
<td>StartDate</td>
<td>The starting date of the campaign.</td>
</tr>
<tr>
<td>EndDate</td>
<td>The ending date of the campaign.</td>
</tr>
</tbody>
</table>

**Note:** The system provided attributes automatically track the creation date and creator of the MediaResource.

**Campaign**

The item type *Campaign* is classified as an item. It is implemented as a folder that contains all the advertisements that are used as part of any particular campaign. For example, a campaign item with the name of Campaign X is created, using the semantic type of folder. Any advertisement that is created for this campaign is automatically put into this folder, using auto-linking. This is described further in “Advertisement” on page 328.

Table 13-8 lists the attributes of the Campaign item type.

**Advertisement**

The item type *Advertisement* uses the Document item type classification. It has two allowed document parts:

- **ICMBASE**: This is used to store the advertisement and has the flexibility to be used for any type of file.

- **ICMBASETEXT**: This is used to store the advertisement transcript if it is required. It is only used for text files, and is enabled for text-searching.
Table 13-9 lists the attributes of the Advertisement item type.

**Table 13-9  Advertisement attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CampaignID</td>
<td>The unique identifier of the campaign that this advertisement belongs to. This is a foreign key linked to the Campaign item type. Auto-linking is also used. This means that each advertisement is automatically placed into the appropriate campaign folder.</td>
</tr>
<tr>
<td>AdvertisementID</td>
<td>The identifier of the advertisement.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the advertisement.</td>
</tr>
<tr>
<td>Description</td>
<td>A description of the advertisement.</td>
</tr>
<tr>
<td>StartDate</td>
<td>The starting date of the advertisement.</td>
</tr>
<tr>
<td>EndDate</td>
<td>The ending date of the advertisement.</td>
</tr>
<tr>
<td>StatusID</td>
<td>The identifier of the advertisement’s status. This is a foreign key, linked to the Keyword item type.</td>
</tr>
<tr>
<td>MediaTypeID</td>
<td>The identifier of the advertisement’s media type. This is a foreign key, linked to the Keyword item type.</td>
</tr>
</tbody>
</table>

In addition, an advertisement has three child components as described in Table 13-10.

**Table 13-10  Advertisement child components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaResources</td>
<td>MediaResrceRef - A reference to a media resource item.</td>
</tr>
<tr>
<td>Contributors</td>
<td>ContrTypeID - The identifier of the contributor type. This is a foreign key to the Keyword item type.</td>
</tr>
<tr>
<td></td>
<td>ContributorID - The identifier of the contributor. This is a foreign key to the Employee item type.</td>
</tr>
<tr>
<td></td>
<td>Description - A description of the contribution made.</td>
</tr>
<tr>
<td>Products</td>
<td>ProductID - The identifier of the product. This is a foreign key to the Product item type.</td>
</tr>
</tbody>
</table>

An advertisement can have multiple child components for each child component type. This means that there can be multiple media resources, multiple contributors and multiple products for one advertisement.
13.3.3 Workflow

ACME Marketing requires a simple workflow for advertisements. This workflow goes through a quality assurance review and a legal review, as shown in Figure 13-4.

![Workflow Diagram]

Figure 13-4 Advertisement workflow

A number of system exits are used to assist the workflow process. These are explained in Table 13-11.

Table 13-11 Workflow system exits

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA Review (Reject)</td>
<td>This exit is used to notify the author about the quality assurance review rejection. The author needs to make the appropriate changes and re-submit (re-start) the advertisement on the workflow.</td>
</tr>
<tr>
<td>Legal Review (Reject)</td>
<td>This exit is used to notify the author about the legal assurance review rejection. The author needs to make the appropriate changes and re-submit (re-start) the advertisement on the workflow.</td>
</tr>
</tbody>
</table>
To assist the workflow, the following work nodes are required:

- Quality Assurance Review
- Legal Review

Since different groups of people need to look at different views, two worklists are required, one for looking solely at each work node:

- QA Review Worklist (including QA Review work node only)
- Legal Review Worklist (including Legal Review work node only)

### 13.3.4 Migration and document life-cycle

ACME Marketing would like to replicate all the media resource items and all the approved advertisements to the backup Resource Manager. Additionally, ACME would like to migrate them to TSM based on their types:

- **Media resources:** These are retained on the Resource Manager’s managed disks for three years; afterwards, they are migrated to TSM.

- **Approved advertisements:** (These are approved by legal reviewers). They are retained on the Resource Manager’s managed disks for one year; afterwards, they are migrated to TSM.

To enable this, migration policies are created on each primary Resource Manager as described in Table 13-12.

<table>
<thead>
<tr>
<th>Policy name</th>
<th>Storage class</th>
<th>Retention period</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGTCLASS</td>
<td>FIXED</td>
<td>Forever</td>
</tr>
<tr>
<td>ThreeYearsFixedThenTSM</td>
<td>FIXED; TSM</td>
<td>1095 days; Forever</td>
</tr>
<tr>
<td>OneYearFixedThenTSM</td>
<td>FIXED; TSM</td>
<td>365 days; Forever</td>
</tr>
</tbody>
</table>
Collections are created on each primary server as described in Table 13-13.

**Table 13-13  Collections**

<table>
<thead>
<tr>
<th>Collection name</th>
<th>Migration policy</th>
<th>Storage group</th>
<th>Replication enabled?</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;LOC&gt;.CLLCT001</td>
<td>MGTCLASS</td>
<td>Group1</td>
<td>No</td>
</tr>
<tr>
<td>&lt;LOC&gt;.CLLCT002</td>
<td>ThreeYearsFixedThenTSM</td>
<td>Group1</td>
<td>Yes(^b)</td>
</tr>
<tr>
<td>&lt;LOC&gt;.CLLCT003</td>
<td>OneYearFixedThenTSM</td>
<td>Group1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

a. <LOC> is the three letter code used for each location.

b. Replication, if enabled, is with the Resource Manager NYCRMDB2 and collection NYC.CLLCT.BAK.

All media resources are stored directly into the ThreeYearsFixedThenTSM collection. Advertisements are stored by default into MGTCLASS. The final approval by legal reviewers in the workflow as described earlier triggers a user exit. This exit is responsible for moving the advertisements from the MGTCLASS collection to the OneYearFixedThenTSM collection, thus enabling the one-year migration to TSM and replication to the backup Resource Manager.

### 13.3.5 Security

ACME Marketing would like to limit the creation of the media resources and advertisements to specific groups of users. All users who are not defined in these creators groups should be given READ-ONLY access. In addition, only specific sets of users should be allowed to perform quality assurance and legal reviews. Another group of users should be granted full administrative privileges.

To enable these requirements, two new privilege sets need to be created:

- **ClientUserReadWorkflow**, which includes:
  - All privileges that are included in the ClientUserReadOnly privilege group
  - All privileges in the ClientTaskDocRouting privilege group

- **ClientUserAllPrivsNoDelete**, which includes:
  - All privileges that are included in the ClientUserAllPrivs privilege group, except for these:
    - ClientDeleteBasePart
    - ItemDelete
In addition, user groups need to be created as specified in Table 13-14.

**Table 13-14  User groups**

<table>
<thead>
<tr>
<th>Group name</th>
<th>Privilege set(^a)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD_CREATORS</td>
<td>ClientUserAllPrivs</td>
<td>All users who can create new advertisements.</td>
</tr>
<tr>
<td>MR_CREATORS</td>
<td>ClientUserAllPrivs</td>
<td>All users who can create new media resources.</td>
</tr>
<tr>
<td>QA_APPROVERS</td>
<td>ClientUserAllPrivs</td>
<td>All users who can perform quality assurance approval</td>
</tr>
<tr>
<td>LGL_APPROVERS</td>
<td>ClientUserAllPrivs</td>
<td>All users who can perform legal approval.</td>
</tr>
<tr>
<td>ACME_ADMIN</td>
<td>SysAdminSuper</td>
<td>All Content Manager system administrators</td>
</tr>
<tr>
<td>ACME_READERS</td>
<td>ClientUserReadOnly</td>
<td>Default group to be used by all users who do not belong to any of the defined groups.</td>
</tr>
<tr>
<td>ACME_SUPERUSER</td>
<td>ClientUserAllPrivs</td>
<td>All users who are can create new campaigns, add companies, brands and products and administer keywords.</td>
</tr>
</tbody>
</table>

\(^a\) The privilege set is granted to users individually, in the user definition.

The access control lists are required as specified in Table 13-15.

**Table 13-15  ACLs**

<table>
<thead>
<tr>
<th>ACL name</th>
<th>Users / Groups</th>
<th>Description</th>
</tr>
</thead>
</table>
| AdvertisementACL | AD_CREATORS \((ClientUserAllPrivsNoDelete)\)  
|                  | QA_APPROVERS \((ClientUserReadOnly)\)                        | Used to control access to the Advertisement item type. |
|                  | LGL_APPROVERS \((ClientUserReadOnly)\)                        |                                                       |
|                  | ACME_READERS \((ClientUserReadOnly)\)                        |                                                       |
| MediaResourceACL | MR_CREATORS \((ClientUserAllPrivsNoDelete)\)  
|                  | ACME_READERS \((ClientUserReadOnly)\)                        | Used to control access to the MediaResource item type. |
13.3.6 Client

ACME Marketing requires a customized client. The data model that has been designed cannot be supported by the standard Content Manager clients due to its extensive use of items, resource items, foreign keys, and references. If ACME would like to use the existing clients, they would need to modify the data model so that these parts of the functionalities are not included. Having said this, the Content Manager Client for Windows now gives you the ability to display any attributes specified as a foreign-key in a drop-down list for selection by the user. This is excellent to prevent users from typing in an invalid key. This involves the classification of all item types as Documents, despite the fact that many will have no objects associated with them at all.

Also, the normalized nature of the data model would have to be changed, so that the users would have to enter all information (for example product and customer information) for each advertisement they create, rather than being able to select from a list. This could lead to a large amount of redundant and messy data as product information is repeated for each product, and no constraints are made, meaning that the same logical product could be recorded in any number of ways. Due to these reasons, ACME Marketing has decided to pursue a customized client, allowing them to keep the previously design data model and exposing the complete set of functionality of the Content Manager Library Server.

ACME Marketing has decided to design and develop a Web-based client. The reasons behind this decision are as follows:

<table>
<thead>
<tr>
<th>ACL name</th>
<th>Users / Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QAApprovalACL</td>
<td>QA_APPROVERS (ClientUserReadWorkflow)</td>
<td>Used to control access to the QA Review step of the workflow and the associated worklist.</td>
</tr>
<tr>
<td>LGLApprovalACL</td>
<td>LGL_APPROVERS (ClientUserReadWorkflow)</td>
<td>Used to control access to the Legal Review step of the workflow and the associated worklist.</td>
</tr>
<tr>
<td>ConfigItemsACL</td>
<td>ACME_SUPER_USERS (ClientUserAllPrivs) ACME_READERS (ClientUserReadOnly)</td>
<td>Assigned to all other item types. All standard users have READ-ONLY access to these items. The items such as keywords and products are created by super users.</td>
</tr>
</tbody>
</table>
Ease of deployment: Rather than building custom clients that need to be deployed to each user, a Web-based application can be immediately used by all users with an Internet browser.

Use of available infrastructure: Since WebSphere Application Server will be located at each site, ACME Marketing would like to use these servers to host the new Web application in addition to the Resource Managers.

The customized solution requires the components described in Table 13-16.

**Table 13-16  Client components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| System administration      | This component should enable an administrator to:  
  - Create new users (system users and employee items), assigning them with the appropriate privilege sets and the appropriate user group(s).  
  - Perform general administration tasks such as deletion of items when requested.  
  Note: The administration component does not replace the System Administration Client; it only assists with the creation of the users and the maintenance of the items in relation to the data model. |
| Application administration | This component should enable the application super user to:  
  - Create new campaigns. (Note, these are created using the Folder semantic type so that they contain advertisements.)  
  - Create and maintain lists of:  
    - Customers  
    - Brands  
    - Products  
    - Keywords |
| Content creation           | This component allows users to create and view content, (either media resources or advertisements,) and submit advertisements for approval if required.  
  All creations are assisted with the provision of drop-down lists.  
  The values for these lists are populated by the application administrators when they populate the lists of campaigns, customers, brands, products and, most importantly, keywords.  
  For example, when creating an advertisement, the creator selects what campaign the advertisement belongs to from a drop-down list of all the items of type Campaign.  
  Note: Whenever an attribute is marked as a foreign key in the data model, the application provides a drop-down list to enable a user to select a valid value that is represented by an underlying ID. |
13.4 Implementing the solution

Now that the solution’s design has been made, we need to implement the solution. To start with, the Content Manager system needs to be installed and configured.

13.4.1 System installation and configuration

This section includes server installation and configuration.

**Note:** All servers are AIX 5.2. The versions and fixpacks are for this case study only. The versions may be different, depending on your situation.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
<td>This component allows users to:</td>
</tr>
<tr>
<td></td>
<td>- Open worklists to see items requiring reviews.</td>
</tr>
<tr>
<td></td>
<td>- Approve or reject items.</td>
</tr>
</tbody>
</table>

**Note:** For this case study, we do not discuss J2EE application design and architecture since it is beyond the scope of this redbook. There are numerous WebSphere redbooks completely dedicated to this topic. For more information on application development for Content Manager, see Chapter 6, “Application development overview” on page 131.

*Installing servers*

We need to perform the following server installations at each location:

- Install servers in New York. This includes the installation of:
  - **Library Server:** ICMNLSDB
    
    The following components should be installed on the server:
    - IBM DB2 Universal Database Enterprise Edition Version 8.2
      (This is equivalent to DB2 8.1 FP 10)
    - IBM DB2 Universal Database Enterprise Edition Net Search Extender (NSE) Version 8.2
    - IBM Content Manager Library Server Version 8.3
  - **Resource Manager:** NYCRMDB1
    
    The following components should be installed on the server:
• IBM DB2 Universal Database Enterprise Edition Version 8.2
  (This is equivalent to DB2 8.1 FP 10)
• Tivoli Storage Manager (TSM) API Client Version 5.3
• IBM WebSphere Application Server Version 5.1.1.7 (including
  HTTPServer)
• IBM Content Manager Resource Manager Server Version 8.3
• Information Integrator for Content Version 8.3 (Content Manager
  connectors for use by custom client)
  – **Resource Manager:** NYCRMDB2
    The following components should be installed on the server:
    • IBM DB2 Universal Database Enterprise Edition Version 8.2
      (This is equivalent to DB2 8.1 FP 10)
    • IBM WebSphere Application Server Version 5.1.1.7 (including
      HTTPServer)
    • IBM Content Manager Resource Manager Server Version 8.3
  – Tivoli Storage Manager (TSM)
    The following component needs to be installed on the server:
    • IBM Tivoli Storage Manager Version 5.3

  ➤ Install servers in Sydney. This includes the installation of:
    – **Resource Manager:** SYDRMDB1
      The following components should be installed on the server:
      • IBM DB2 Universal Database Enterprise Edition Version 8.2
        (This is equivalent to DB2 8.1 FP 10)
      • Tivoli Storage Manager (TSM) API Client Version 5.3
      • IBM WebSphere Application Server Version 5.1.1.7 (including
        HTTPServer)
      • IBM Content Manager Resource Manager Server Version 8.3
      • Information Integrator for Content Version 8.3 (Content Manager
        connectors for use by custom client)

  ➤ Install servers in London. This includes the installation of:
    – **Resource Manager:** LONRMDB1
      The following components should be installed on the server:
      • IBM DB2 Universal Database Enterprise Edition Version 8.2
        (This is equivalent to DB2 8.1 FP 10)
Content Manager Implementation and Migration Cookbook

- Tivoli Storage Manager (TSM) API Client Version 5.3
- IBM WebSphere Application Server Version 5.1.1.7 (including HTTPServer)
- IBM Content Manager Resource Manager Server Version 8.3
- Information Integrator for Content Version 8.3 (Content Manager connectors for use by custom client)

Configuring servers
We need to perform the following server configurations:

- Configure Library Server:
  We registered four Resource Managers with the Library Server, enabling LAN Cache for all Resource Managers.
  We also modify the Library Server's Default Storage Options. We set the default Resource Manager and default collection to be retrieved from User.

- Configure each primary Resource Manager by performing the following steps:
  a. Create the following storage classes:
     - FIXED: The default storage class using the JFS device manager
     - TSM: The storage class for TSM using the ICMADDM device manager
  b. Define the TSM Server to the Resource Manager.
  c. Create a new TSM volume using:
     TSM Management Class: TSMMC
     Server name: The name of the server defined in the previous step.
     Storage class: TSM
     Assignment: Assigned to Group01
  d. Create the following migration policies:
     - MGTCLASS
       Storage class: FIXED, Retention period: Forever
     - ThreeYearsFixedThenTSM
       Storage class: FIXED, Retention period: 1095 days
       Storage class: TSM, Retention period: Forever
     - OneYearFixedThenTSM
       Storage class: FIXED, Retention period: 365 days
       Storage class: TSM, Retention period: Forever
e. Define each of the other Resource Managers (including the backup) to the current Resource Manager. For example, on NYCRMDB1, define SYDRMDB1, LONRMDB1 and NYCRMDB2. Note, these remote databases are also cataloged using the DB2 Client Configuration Assistant.

f. Create the following workstation collections (note, <LOC> is the three letter code used for each location):

- <LOC>.CLLCT001
  Migration policy: MGTCLASS
  Storage group: Group01
  Replication with: NONE
- <LOC>.CLLCT002
  Migration policy: ThreeYearsFixedThenTSM
  Storage group: Group01
  Replication with: NYCRMDB2 (NYC.CLLCT.BAK)
- <LOC>.CLLCT003
  Migration policy: OneYearFixedThenTSM
  Storage group: Group01
  Replication: NYCRMDB2 (NYC.CLLCT.BAK)

Configure the backup Resource Manager (NYCRMDB2) as follows:

a. Create the following storage class:
   FIXED: The default storage class using the JFS device manager.

b. Create the following migration policy:
   MGTCLASS
   - Storage class: FIXED, Retention period: Forever

c. Register each of the other Resource Managers.

d. Create the following workstation collection:
   NYC.CLLCT.BAK
   - Migration policy: MGTCLASS
   - Storage group: Group01
   - Replication with: NONE
**Configure Tivoli Storage Manager**

The setup of the policy definitions for TSM is outside the scope of this redbook. Typically, an object needs to migrate from Content Manager to a TSM disk pool. At a later specified date, the TSM server is then responsible to migrate the content to another form of media, such as tape storage pool.

### 13.4.2 Workflow

To implement the workflow, we create the work nodes displayed in Table 13-17. Note that they are all work baskets. The Overload limit is set to 0 for all.

**Table 13-17 Work nodes (baskets)**

<table>
<thead>
<tr>
<th>Name</th>
<th>ACL</th>
<th>Exits</th>
</tr>
</thead>
<tbody>
<tr>
<td>QualityAssuranceReview</td>
<td>QAApprovalACL</td>
<td>QAResReview (Reject): This exit is used to notify the author about the quality assurance review rejection. The author needs to make the appropriate changes and re-submit the advertisement on the workflow.</td>
</tr>
<tr>
<td>LegalReview</td>
<td>LGLApprovalACL</td>
<td>LegalReview (Reject): This exit is used to notify the author about the legal assurance review rejection. The author needs to make the appropriate changes and re-submit the advertisement on the workflow. LegalReview (Approve): This exit is used to notify the author about the final approval of the advertisement. It also move the object to another collection of the Resource Manager.</td>
</tr>
</tbody>
</table>
A process is created as shown in Table 13-18 and is given the ACL, AdvertisementACL.

Table 13-18 Advertisement workflow

<table>
<thead>
<tr>
<th>From node</th>
<th>Selection</th>
<th>To node</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Continue</td>
<td>QualityAssuranceReview</td>
</tr>
<tr>
<td>QualityAssuranceReview</td>
<td>Approve</td>
<td>LegalReview</td>
</tr>
<tr>
<td>QualityAssuranceReview</td>
<td>Reject</td>
<td>END</td>
</tr>
<tr>
<td>LegalReview</td>
<td>Approve</td>
<td>END</td>
</tr>
<tr>
<td>LegalReview</td>
<td>Reject</td>
<td>END</td>
</tr>
</tbody>
</table>

13.4.3 Migration and document life-cycle

The migration and document life-cycle requirements are implemented as a part of the implementation described in 13.4.1, “System installation and configuration” on page 336.

13.4.4 Security

We create each user group, privilege set, and ACL as designed and defined in 13.3.5, “Security” on page 332. The administration users are created and granted AllPrvis. One of the application components allows general user creation. Using this application, each user is put into the ACME_READERS group and at least one other group depending on what their intended usage of the system is.
Each user is given the following defaults:

- **ResourceManager**: Their local primary Resource Manager.
- **Collection**: `<LOC>.CLLCT001` where `<LOC>` is the three letter code of their location.
- **Item ACL**: AdvertisementACL.

### 13.4.5 Data model

To implement the data model that they designed, ACME Marketing must go through a number of steps. First they create the following attributes:

- AdvertisementID
- BrandID
- CampaignID
- ContributorID
- ContrTypeID
- CustomerID
- Description
- EmployeeID
- EndDate
- ExpertiseArea
- FirstName
- GroupID
- IndustryTypeID
- KeywordID
- LastName
- MediaResrceID
- MediaResrceRef (Reference Attribute)
- MediaTypeID
- Name
- ProductID
- StartDate
- StatusID
- UserID

With these created, ACME Marketing create the item types in the order shown in Table 13-19. The attributes for each item type are listed in 13.4.5, “Data model” on page 342.

**Note:** All of the foreign keys are defined with No action for the update rule and Restrict for the delete rule.
<table>
<thead>
<tr>
<th>Name</th>
<th>Foreign key</th>
<th>Other information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>N/A</td>
<td>All attributes are mandatory. Description represents item.</td>
</tr>
<tr>
<td>Employee</td>
<td>EmployeeID → External database</td>
<td>All attributes are mandatory. UserID represents item.</td>
</tr>
<tr>
<td>Customer</td>
<td>IndustryTypeID → KeywordID (Keyword)</td>
<td>All attributes are mandatory. Name represents item.</td>
</tr>
<tr>
<td>Brand</td>
<td>CustomerID → CustomerID (Customer)</td>
<td>All attributes are mandatory. Name represents item.</td>
</tr>
<tr>
<td>Product</td>
<td>BrandID → BrandID (Brand)</td>
<td>All attributes are mandatory. Name represents item.</td>
</tr>
<tr>
<td>Campaign</td>
<td>CustomerID → CustomerID (Customer)</td>
<td>All attributes are mandatory. Name represents item.</td>
</tr>
<tr>
<td>MediaResource</td>
<td>MediaTypeID → KeywordID (Keyword)</td>
<td>All attributes are mandatory. Description represents item. Media object class: DLKlobICM</td>
</tr>
<tr>
<td></td>
<td>StatusID → KeywordID (Keyword)</td>
<td>Default storage: NYCRMDB1 (NYC.CLLCT001)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACL: MediaResourceACL</td>
</tr>
<tr>
<td>Advertisement</td>
<td>CampaignID → CampaignID (Campaign)</td>
<td>All attributes are mandatory. Name represents item.</td>
</tr>
<tr>
<td></td>
<td>StatusID → KeywordID (Keyword)</td>
<td>ACL: AdvertisementACL</td>
</tr>
<tr>
<td></td>
<td>MediaTypeID → KeywordID (Keyword)</td>
<td>Auto-linking: enabled to Campaign item type (CampaignID → CampaignID)</td>
</tr>
<tr>
<td></td>
<td>Child Components</td>
<td>- Folder Contains</td>
</tr>
<tr>
<td></td>
<td>ContrTypeID (Contributors) →</td>
<td>Versioning: Always create (limited to 3)</td>
</tr>
<tr>
<td></td>
<td>KeywordID (Keyword)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ContributorID (Contributors) →</td>
<td>Document Parts</td>
</tr>
<tr>
<td></td>
<td>EmployeeID (Employee)</td>
<td>ICMBASE, ICMBASETEXT</td>
</tr>
<tr>
<td></td>
<td>ProductID (Products) → ProductID (Product)</td>
<td>Default storage: NYCRMDB1 (NYC.CLLCT001)</td>
</tr>
</tbody>
</table>
13.4.6 Client

Rather than using the existing Application Server (icmrn), ACME Marketing creates another WebSphere Application Server to run on the same machine. In other words, each primary Resource Manager server has two application servers:

- Resource Manager (icmrn)
- Customized client application (AcmeClient)

Putting these on separate servers (but on the same machine) allows them to have extra flexibility down the track, yet this makes use of the existing infrastructure.

The AcmeClient application server’s classpath was modified to include:

- C:\Program Files\IBM\db2cvm8\lib\cmbcm81.jar
- C:\Program Files\IBM\db2cvm8\lib\cmbicm81.jar
- C:\Program Files\IBM\db2cvm8\lib\cmbsdk81.jar
- C:\Program Files\IBM\db2cvm8\cmgmt\ (This folder is included in the classpath so that cmbcmenv.properties can be located)

As discussed in designing the solution, 13.3.6, “Client” on page 334, the application needs to have a number of components. Each component is developed with the assistance of the sample code that is shipped with the connectors and with using WebSphere Studio Application Designer Version 5.1.
In this part of the book, we discuss Content Manager migration. This includes migration on Multiplatforms, for TSM, and for Content Manager custom applications. In addition, we describe an approach and process for special migration scenarios, such as cross platform migration, and migration from a third-party product.

**Important:** For the most recent updates on Content Manager for z/OS, refer to the existing product manuals and the following redbook:

*DB2 Content Manager for z/OS: Implementation, Installation, and Migration, SG24-6476*
Upgrade and migration on multiplatforms

In this chapter, we address Content Manager system upgrade and migration on multiplatforms. We cover upgrading a Content Manager system from Version 8.2 to Version 8.3 on Windows and AIX. In addition, we cover migration of a Content Manager system from Version 6.1 or Version 7.1 to Version 8.3.
14.1 Introduction

This chapter of the redbook should be used in conjunction with the following IBM manuals:

- *IBM Content Manager for Multiplatforms - Migrating to Content Manager Version 8.3*, SC27-1343
- *IBM Content Manager for Multiplatforms - Planning and Installing Your Content management System Version 8.3*, GC27-1332

We describe various issues to consider when you upgrade or migrate your Content Management (CM) system over the same or different platform.

There are eight supported migration scenarios:

- Scenario 1: Out of the box migration, from CM V6.1 or 7.1 to CM V8.3 (see Table 14-1 for details)
- Scenario 2: From CM V6.1 or 7.1 with VideoCharger to CM V8.3 (see Table 14-2 for details)
- Scenario 3: From CM V6.1 or 7.1 with VisualInfo/Digital Library V2 OS/2® Object Server to CM V8.3 (see Table 14-3 for details)
- Scenario 4: From CM V6.1 or 7.2 with custom folder manager application to CM V8.3 with custom ICM connector application (see Table 14-4 for details)
- Scenario 5: From CM V6.1 or 7.1 with custom DL connector application to CM V8.3 with custom ICM connector application (see Table 14-5 for details)
- Scenario 6: From CM V6.1 or 7.1 with EIP toolkit and custom EIP application to CM V8.3 with Information Integrator for Content APIs and custom application (see Table 14-6 for details)
- Scenario 7: From CM V6.1 or 7.1 with EIP toolkit and eClient to CM V8.3 with Information Integrator for Content APIs and eClient (see Table 14-7 for details)
- Scenario 8: From CM V7.1 to system with both CM V7.1 and V8.3 (see Table 14-8 for details)
### Table 14-1  Scenario 1: Out of the box migration

<table>
<thead>
<tr>
<th>Original Configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT® or AIX, or VisualInfo™ or Digital Library V2.4 Library Server on OS/2</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>CM V6.1 or V7.1 Client for Windows or V2.4 Client for OS/2</td>
<td>V8.3 Client for Windows</td>
</tr>
</tbody>
</table>

### Table 14-2  Scenario 2: CM V6.1 or 7.1 with VideoCharger to V8.3

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>VideoCharger V7.1 on Windows NT or AIX</td>
<td>VideoCharger V8.3 on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Client for Windows</td>
<td>V8.3 Client for Windows</td>
</tr>
</tbody>
</table>

### Table 14-3  Scenario 3: CM V6.1 or 7.1 with VisualInfo/Digital Library V2 OS/2 Obj Svr

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>VisualInfo or Digital Library V2 Object Server on OS/2</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Client for Windows</td>
<td>V8.3 Client for Windows</td>
</tr>
</tbody>
</table>

### Table 14-4  Scenario 4: CM V6.1 or 7.2 with custom folder manager application to DB2

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Custom folder manager application</td>
<td>Custom ICM connector application</td>
</tr>
</tbody>
</table>
### Table 14-5  Scenario 5: CM V6.1 or 7.1 with custom DL connector application

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Custom DL connector application</td>
<td>Custom ICM connector application</td>
</tr>
</tbody>
</table>

### Table 14-6  Scenario 6: CM V6.1 or 7.1 with EIP toolkit and custom EIP application

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Information Integrator for Content V7.1 Toolkit</td>
<td>Information Integrator for Content V8.3 connector APIs</td>
</tr>
<tr>
<td>Custom federated application using Information Integrator for Content V7</td>
<td>Custom federated application using Information Integrator for Content V8.3</td>
</tr>
</tbody>
</table>

### Table 14-7  Scenario 7: CM V6.1 or 7.1 with EIP toolkit and eClient

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier CM Library Server on Windows NT or AIX</td>
<td>V8.3 Library Server on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Earlier CM Object Server on Windows NT or AIX</td>
<td>V8.3 Resource Manager on Windows 2000 or AIX</td>
</tr>
<tr>
<td>Information Integrator for Content V7 Toolkit</td>
<td>DB2 Information Integrator for Content V8.3 connector APIs</td>
</tr>
<tr>
<td>Information Integrator for Content V7 eClient</td>
<td>DB2 Information Integrator for Content V8.3 eClient</td>
</tr>
</tbody>
</table>
Table 14-8  Scenario 8: CM V7.1 to system with both V7.1 and 8.3

<table>
<thead>
<tr>
<th>Original configuration</th>
<th>Destination configuration</th>
</tr>
</thead>
</table>
| CM V7.1 Library Server on Windows NT or AIX | The following, coexisting Library Servers:  
  - V7.1 Library Server on Windows 2000 or AIX  
  - V8.3 Library Server on Windows 2000 or AIX |
| CM V7.1 Object Server on Windows NT or AIX | The following, coexisting Resource Managers:  
  - V7.1 Object Server on Windows 2000 or AIX  
  - V8.3 Resource Manager on Windows 2000 or AIX |
| CM V7.1 Client for Windows | The following, coexisting clients:  
  - V7.1 Client for Windows  
  - V8.3 Client for Windows  
  - Information Integrator for Content V8.3 federated application: eClient or custom |

With Content Manager V8.3, you can migrate from Object Server to Resource Manager within the same platform or across different platforms (such as from Windows to AIX, Sun, Linux, and vice versa). Earlier versions of Content Manager (Version 6.1 and Version 7), VisualInfo, or Digital Library Version 2 Object Server are also supported.

**Important:** When migrating VisualInfo or Digital Library Version 2 Object Server, you must first migrate your objects to Content Manager Version 6.1 or Version 7.1 Object Server.

The Version 8.3 Library Server can be on the same or on a different machine as the Library Server of the earlier Content Manager system. If the V8.3 Library Server is on the same machine as the earlier Library Server, make sure that you use a different name and install path for the Version 8.3 Library Server database.

In order for any migration to succeed, your Content Manager Version 8.3 system must have at least as many Resource Managers as there are Object Servers in your Content Manager Version 7.1 system. This is so, regardless of whether you are planning to move your Resource Manager(s) to a different physical machine or not, during the migration.
Notice that it is not possible to migrate from a Content Manager Version 7.1 system to Content Manager Version 8.3 on Sun Solaris; this is because Sun Solaris was not supported as a platform for Content Manager Version 7.1; consequently, no Sun Solaris migration tools have been written. However, now you can upgrade Content Manager Version 8.2 to Content Manager Version 8.3 on Sun Solaris.

### 14.2 Upgrade considerations

If you have a Content Manager Version 8.2 for Multiplatforms, you can upgrade the following components to Content Manager Enterprise Edition Version 8.3:

- IBM DB2 Content Manager Server Version 8.2 to Content Manager Enterprise Edition Version 8.3.
- IBM DB2 Information Integrator for Content Version 8.2 (formerly, Enterprise Information Portal - EIP) to Information Integrator for Content Version 8.3 (as of Version 8.3, it is part of the Content Manager components).
- Content Manager eClient Version 8.2 to eClient Version 8.3.
- IBM DB2 Content Manager Windows Client Version 8.2 to Windows Client Version 8.3.

**Important:** When you upgrade Content Manager components, upgrade them following the same order in the list mentioned above.

### 14.2.1 General considerations

Before you start the upgrade process, we recommend to do the following tasks:

- Make a backup of the Library Server and Resource Manager databases.
- Apply the DB2 Content Manager Fix Pack 8 to Version 8.2 or later for the components that you want to upgrade.

**Important:** If you are planning to use a Resource Manager Version 8.2 with a Library Server Version 8.3, then you must apply the Fix Pack 9 or later to the Resource Manager. This scenario applies when you have two or more distributed DB2 Content Manager systems with different versions sharing the same Resource Manager server.

- Ensure that all the prerequisite products meet the requirements for Content Manager Version 8.3. You can refer to Chapter 5 of *IBM Content Manager for Multiplatforms - Planning and Installing Your Content management System Version 8.3*, GC27-1332.
Important: If you want to upgrade Content Manager eClient Version 8.2 to Version 8.3, you must install WebSphere Application Server Version 5.1 with Fix Pack 1. You can migrate eClient Version 8.2 during the WebSphere Application Server Version 5.1 installation process, or later using the migration tool provided by the WebSphere Application Server.

Another alternative is uninstall the eClient Version 8.2 from WebSphere Application Version 5.0, and then install WebSphere Application Server Version 5.1. When WebSphere Application Server installation process completes, install eClient Version 8.3. Please be careful with this option, if you have customized your eClient interface.

- If the Content Manager Server Version 8.2 and WebSphere information Integrator for Content Version 8.2 are on the same machine, you must upgrade both. The reason is that they share common files, and you cannot upgrade only one of them.
- Ensure that your system is not running before you upgrade.
- Restart your DB2 instance and your application server.
- The upgrade is executed by the same setup process, so the installation program of Content Manager Version 8.3 detects that you are performing an upgrade after the Installation Destination window displays (Figure 14-1).

![DB2 Content Manager Enterprise Edition V8.3 Installation](image)

*Figure 14-1  DB2 Content Manager Version 8.3 upgrade detection*
When you are upgrading Information Integrator for Content Version 8.3, the upgrade process only performs an upgrade of the current set of components of Version 8.2. You will not have the opportunity to select additional components, such as Web services, to install. If you want to install additional components, you must run the Information Integrator for Content installation program for Version 8.3 again.

Some connectors supported in Version 8.2 will not be supported in Version 8.3. Therefore, a warning window displays before you upgrade, indicating which connectors are not available after upgrade (Figure 14-2).

Remember that, after the upgrade, you cannot return to the previous version.

The upgrade of Content Manager Client for Windows Version 8.3 is similar to a normal installation. The installation program detects a Version 8.2 and automatically uninstalls this version and installs Version 8.3.

The sample data from Content Manager Version 8.2 should be unloaded before the upgrading to DB2 Content Manager Version 8.3. Unloading this data allows Content Manager Version 8.3’s First Steps utility to load and unload the sample data properly.
14.2.2 UNIX considerations

In a UNIX environment, there are modifications related to user and group IDs, configuration files, and DB2 that you need to consider.

**User and group IDs**

The user ID requirements have changed in Version 8.3. You need to perform the following steps related to user and group IDs before upgrading your system to Version 8.3:

1. Create a new Content Manager administration group, ibmcmgrp.
2. Create a new Content Manager administration user ID, ibmcmadm. The primary group for this user ID must be ibmcmgrp.
3. Set the permissions of the home directory of the Content Manager administration user to 775.
4. All the other Content Manager user IDs, such as icmadmin, rmadmin, and icmconct, must be set to be members of the Content Manager administration group, ibmcmgrp.
5. Any additional users who run product commands must be added to the Content Manager administration group. For example, users who run the connector samples need to be part of the administration group.

**Tip:** Make ibmcmgrp the secondary group for these user IDs.

**Configuration files**

You need to add some lines in the .profile file (or to the .bashrc file for Linux) for the ibmcmadm, icmadmin, rmadmin, and icmconct user IDs:

```
IBMCMROOT=/opt/IBM/db2cmv8
export IBMCMROOT
```

If any of the user profiles has the environment variables ICMROOT, CMBROOT, or CMCOMMON defined, modify them as follows:

```
ICMROOT=$IBMCMROOT
CMBROOT=$IBMCMROOT
CMCOMMON=$IBMCMROOT/cmgmt
```

**Tip:** If you are certain that you do not have custom code that refers to ICMROOT, CMBROOT or CMCOMMON, then you can remove these variables from your .profile file. The Version 8.3 code does not use these variables.
Ensure that the following line appears in the .profile file (or the .bashrc file for Linux) in the home directory of the icmadmin and rmadmin user IDs:

```
. DB2INSTHOME/sqllib/db2profile
```

Where $DB2INSTHOME$ is the home directory of the DB2 instance, such as /home/db2inst1.

**Note:** There is a space between the period (.) and $DB2INSTHOME$.

To ensure that the correct commands are used, update PATH in the .profile file (or the .bashrc file for Linux) for the icmadmin, rmadmin, and icmconct user IDs:

```
PATH=$IBMCMROOT/java/jre/bin:/usr/bin:$IBMCMROOT/bin:$PATH
export PATH
```

Correct the following two lines in DB2INSTHOME/sqllib/profile.env (for example, /home/db2inst1/sqllib/profile.env on AIX) to look like this:

```
DB2LIBPATH=/usr/lib:/opt/IBM/db2cmv8/lib
DB2ENVLIST='LD_LIBRARY_PATH LIBPATH IBMCMROOT ICMDLL EXTSHM'
```

Create or modify DB2INSTHOME/sqllib/userprofile (for example, /home/db2inst1/sqllib/userprofile on AIX) to contain the following data:

```
IBMCMROOT=/opt/IBM/db2cm8
ICMDLL=/home/db2fenc1
LIBPATH=$IBMCMROOT/lib:$LIBPATH
export IBMCMROOT
export ICMDLL
export LIBPATH
```

**DB2 requirements**

To ensure that the new environment is used for future operations, restart DB2 UDB for the Library Server administration ID:

```
db2stop (for stopping DB2 UDB)
db2start (for starting DB2 UDB)
```

If dbstop fails, enter db2stop force.

If your Resource Manager resides in a different DB2 instance, then you should also restart DB2 UDB for Resource Manager administration ID.

For DB2 Content Manager Version 8.3 products, there is no requirement to define any DB2 Content Manager product related environment variables in the .profile file (or the .bashrc file for Linux) of the root user ID. So, if it exists, the following information should be removed from the .profile file of the root user ID:
- `ICMROOT` and any reference to the `/usr/lpp/icm` value on the environment variable in `PATH`, `CLASSPATH` and `LIBPATH`
- `CMBROOT` and any reference to the `/usr/lpp/cmb` in `PATH`, `CLASSPATH` and `LIBPATH`
- Remove the line containing `db2profile: . DB2INSTHOME/sqlib/db2profile`.

To summarize, the following files (or equivalent files, depending on your operating system) should be updated:

```
/home/icmcmadm/.profile
/home/icmadmin/.profile
/home/rmadmin/.profile
/home/icmconct/.profile
/home/db2inst1/sqlib/profile.env
/home/db2inst1/userprofile
```

After updating the files, restart your DB2 instance.

**Important:** If you have a system with both Content Manager and Information Integrator for Content Version 8.2 installed on the same server or separated servers, upgrade DB2 Content Manager first.

### 14.3 Migration considerations

There are several issues to consider when planning and performing a migration:

- What is new in the migration tool for Content Manager V8.3
- Disk space
- Backup
- Timing
- Migrating Content Manager systems earlier than Version 7

#### 14.3.1 What is new in the migration tool for Content Manager V8.3

The migration tool helps you to migrate from Content Manager Version 6.1 or Version 7.1 to Content Manager Version 8.3 on Windows and AIX. This migration utility is now available for download at:


The migration utility is not shipped with the Content Manager Version 8.3 product images, nor in the Content Manager Version 8.3 Fix Packs. Instead, the migration utility is maintained separately, as updated versions of the migration utility become available. To obtain updated versions, visit the URL mentioned above and download the utility.
The Content Manager Version 8.3 migration utility has the following improvements:

- It supports Content Manager Version 8.3 for Oracle.
- Auto-linking metadata is migrated.
- Internal changes make it possible to use Content Manager Version 8.2 and Version 8.3 C++ APIs.

The migration utility works the same way as it did for Content Manager Version 7.1 to Content Manager Version 8.2 migrations, with GUI and commands to export data from the Version 7.1 system and import it into the target Content Manager Version 8 system.

Migrating to Content Manager Version 8 (also referred to as the Migration Guide) has been improved with added screen shots, flow chart, and restartability points. Content Manager Version 8.3 Migration Guide is available at:


14.3.2 Disk space

The disk space requirements fall into the following categories:

- Disk space required for the migration process
- Disk space required for the new, migrated system

Both areas are important to cover before any actual migration is started.

**Disk space required for the migration process**

When migrating the Content Manager Version 6.1 or Version 7.1 Object Server database, a temporary DB2 table, CM2ICMPARTS, is created by the Migration Wizard in the Content Manager Version 6.1 or Version 7.1 Object Server database. This table is used to build the data that is migrated to the Content Manager Version 8.3 Resource Manager database. The size of the temporary table will be approximately equal to the current size of your Content Manager Version 6.1 or Version 7.1 Object Server database. So you must make sure that you have at least more than 50% free space in the Content Manager Version 6.1 or Version 7.1 Object Server database before you start the migration process.

The Migration Wizard also creates work files during the export of the Content Manager Version 6.1 or Version 7.1 Library Server and Object Server data. The total size of these export files corresponds directly to the size of your current database.
Disk space required for the new, migrated system

There is no accurate way to calculate the space needed for the new Content Manager Version 8.3 Library Server database, because the data model has changed significantly to support extended features in the new version. Some data types such as multi-value attribute key fields may expand into multiple definitions, whereas others remain the same. Therefore, the space requirement is dependent on the actual data in your current system.

As a rule of thumb, the new database size should be three to four times the size of your existing database.

The size of the Content Manager Version 8.3 Resource Manager database will be approximately the same as your current Content Manager Version 6.1 or Version 7.1 Object Server database.

Calculating your disk space needs

If the disk space in your current system and the target system is limited, we recommend that you take extra time to calculate the disk space needed for the migration and future operation. If you do not have enough disk space, this is a good time to move to a new, larger system or expand your existing system.

14.3.3 Backup

Remember to back up your system if you plan to make any changes to it. This is especially important when planning a migration. When you plan for the Content Manager migration, be sure to include sufficient time to take backups, and, if necessary, restore them. Also be sure that you test the integrity of your backup. For large databases, the time and space requirements can be considerable.

We assume that your enterprise already has stringent backup procedures in place for your databases and other data, and that the system administrator regularly tests restoration procedures. This section should therefore serve as a reminder to your system administrator.

The following components must be backed up:

- Library Server database
- Object Server database
- LBOSDATA directory

Keep in mind that a backup of your system must be consistent across all of the components to ensure that the Library Server and the Object Server databases are synchronized.
After performing a backup of the databases, we recommend that you optimize all of them prior to migration. This increases performance. See 18.2, “Optimizing server databases” on page 472 for instructions on how to do this.

14.3.4 Timing

When creating a project plan for your Content Manager migration project, there is an acceptable time frame in which you must complete all your tasks. You need to consider:

- The time involved in clearing the Object Server staging area
- The time involved in incremental or full system and data backup
- The time and effort involved in running the migration on a test system
- The time involved in installing or removing the products
- The effort involved in migrating each component of your system
- The order in which the components must be migrated
- The restoration time in case the migration fails
- The time involved in testing and validating the newly migrated system

14.3.5 Migrating Content Manager systems earlier than Version 7.1

In order to migrate to Content Manager Version 8.3, the existing system must be Content Manager Version 6.1 or Content Manager Version 7.1. Within this redbook, we base our migration scenario on a Content Manager Version 7.1 system. The processes and procedures described should work equally well on a Content Manager Version 6.1 system.

If you are currently running on a version of Content Manager earlier than Version 6.1, you must first migrate to Version 6.1 or Version 7.1 before you can migrate to Version 8.3. This migration procedure is covered within the documentation for the relevant release level. For example, see Chapter 12, “Migrating a Windows or AIX Content Manager database” from, *IBM Content Manager for Multiplatforms - Planning and Installing Content Manager Version 7.1*, GC27-0864.

14.4 Data migration overview

In this section, we define the data migration steps involved in migrating data from one Content Manager system to another.

There are many Content Manager configurations that may exist within a Version 7.1 system, such as a stand alone environment (the Library Server and the Object Server are on the same machine), a distributed system with one Object
Server (the Library Server and the Object Server are on different physical machines), and a distributed system with more than one Object Server (all servers are on different machines). However, the main steps involved during a migration are the same.

### 14.4.1 Data migration steps

What does it mean to migrate your data? When you migrate your data to Content Manager Version 8.3, you do not migrate the actual data, or objects. You migrate the data in your system that points to those objects and establish the structure that you use for finding and retrieving those objects. You use the provided Migration Wizard to migrate your system definition data (for example, user IDs, access control lists, and index class definitions) and your user data (for example, attribute values, relationships between items such as folder relationships, and checkout status information).

Note that, as a result of changes to the data model, some of your data cannot be migrated one-for-one, because it does not directly map. An example of one-for-one migration is index classes, which migrate to item types. Alternatively, multi-valued attributes do not exist explicitly in Content Manager Version 8.3; if you had them in Content Manager Version 7.1, they are migrated to Content Manager Version 8.3 as child components.

The main activities involved during data migration are as follows:

1. Export definitions and data from you Content Manager Version 7.1 Library Server.
2. Create new definitions on your Content Manager Version 8.3 Library Server.
3. Export user data from you Content Manager Version 7.1 Library Server.
4. Import user data on your Content Manager Version 8.3 Library Server.
5. Export user data from your Content Manager Version 7.1 Object Server(s) and import it on your Content Manager Version 8.3 Resource Manager(s).

**Note:** The migration utilities perform all these steps for you, except for step 6.

### 14.5 General data migration preparation

Regardless of what type of migration you want to perform, there are several common steps you must complete prior to performing the actual migration process. They are as follows:

Chapter 14. Upgrade and migration on multiplatforms 361
1. Validate and document the existing environment.

It is important to ensure that the current Content Manager Version 7.1 system is in a well defined state before running the migration process. We recommend running through a suite of tests to validate that the current system is running as expected. If such a testing suite does not exist, you or the testing team should create one and run through it before any migration takes place. In addition, it is a good idea to ensure that the documentation of the Content Manager system is current and up-to-date. You need a clear understanding of the current system in order to verify that the migration is successful.

2. Back up your system.

3. Complete the necessary installation steps for your environment.

If you are installing Version 8.3 on the same machine as your earlier version of Content Manager, remember to give your Version 8.3 Library Server database a name that is different than your earlier version's Library Server database name; otherwise, you risk overwriting your existing Library Server database. If you use the same name, the Content Manager Version 8.3 installation program will prompt you whether to replace the existing database or not.

4. Download the migration utility.

To migrate your Content Manager Version 7.1 or Content Manager Version 6.1 to DB2 Content Manager Version 8.3 with the latest Fix Pack, you need to get the most recent available Content Manager Version 8.3 downloadable, go to:


5. Ensure that you have the following information:

   – For the earlier version of Content Manager, you need to know:
     • The Library Server name
     • The user ID
     • The password

   – For the new Content Manager Version 8.3 Library Server, you need to know:
     • The Library Server name
     • The administrator user ID
     • The administrator user ID password
     • The schema name
6. Ensure that no users are logged into the system.

   From the Content Manager System Administration Client connecting to your earlier Content Manager system, verify that there are no users logged in to the Content Manager.

7. Complete replication.


   See 14.5.2, “Clearing Object Server staging area” on page 365 to do the following steps:
   a. Destage all objects in the staging area.
   b. Purge the staging area.

9. Shut down all the servers.


10. Perform another incremental backup.

    This is optional. This is to ensure that you have a quick starting point in case the migration process does not complete successfully. Remember, you should have a full backup before installing any new software and fixpack to your existing production system.

14.5.1 Before you begin

Before you begin, consider the following information:

- The migration is not compatible with DB2 Universal Database Version 5.2 run-time executable and will give an error indicating a missing library when you execute the frn2icml command. The migration utility has been built and bound using DB2 Universal Database Version 7.2. If the current version of Content Manager that you are running is on DB2 Universal Database Version 5.2, you must first upgrade to DB2 Universal Database Version 7.2, Fix Pack 10 or higher, before you can migrate to DB2 Content Manager Version 8.3.

Before you migrate from Content Manager Version 7.1 to Content Manager Version 8.3, we recommend that you perform REORGCHK UPDATE STATISTICS against the Version 7.1 Library Server database, then run REORG/RUNSTATS as indicated by REORGCHK.

- If Content Manager Version 8.3 will be running on the same machine as your current version of Content Manager, you must first upgrade your DB2 Universal Database level before you can install DB2 Content Manager Version 8.3 and perform the migration.
If you will be running Content Manager Version 8.3 on a different machine, you should install Content Manager Version 8.3 before the migration. Then, back up the Content Manager Version 7.1 server databases from your DB2 Universal Database Version 5.2 machine and restore these databases on your Content Manager Enterprise Edition Version 8.3 machine.

If you are using Oracle for your database, you must upgrade to Oracle Version 8.1.7.4 (or higher, up to Version 9), or Oracle Version 9i (or Oracle 10) before you begin the migration process.

If you have text indexed data in your previous Content Manager system, you must create your Content Manager Version 8.3 Library Server database with text search support enabled, and make sure that the text search flag is checked in the Library Server configuration through the system administration client. Otherwise, the text search information is lost during migration.

The ICM C++ connector uses the DB2 Universal Database Call Level Interface (CLI) APIs that are included in the DB2 Universal Database runtime client to access DB2 Universal Database. The ICM Java connector uses JDBC™.

The migration utility does not migrate the Library Server configuration from the previous Content Manager system. You must review and modify your Content Manager Version 8.3 Library Server configuration after you install Content Manager Version 8.3 and before the Library Server can be put in production.

Before migrating the Object Server on AIX, you need to manage permissions of the object files so that they can be accessed by both Content Manager Version 7.1 and Content Manager Version 8. In Content Manager Version 7.1, the object files are owned by the AIX user ID that runs the Object Server, that is, osadmin. In Content Manager Version 8, the object files are accessed through the WebSphere user ID. In order to migrate, the object files need to be accessible by both the Version 7.1 AIX user ID and the Version 8.3 AIX user ID. To make the object files accessible by both the Version 7.1 AIX user ID and the Version 8.3 AIX user ID, there are two options:

- Start the Resource Manager Web application server with the osadmin or root user ID.
- Use the `chown` command on the Content Manager volumes. For example, if the Content Manager Version 7.1 AIX Object Server user ID is osadmin, and the Content Manager Version 8.3 AIX WebSphere and Resource Manager user ID is wasadmin, issue the `chown` command to allow the WebSphere and Resource Manager user ID to access the object files:

```
chown -R wasadmin:<WASgroup> <mount_point>/lbosdata
```
In the following sections, we address the tasks of clearing the Object Server staging area and shutting down the servers.

### 14.5.2 Clearing Object Server staging area

The staging area of the Object Server must be cleared before backing up your existing system databases.

**Destage and purge commands**

When new objects are stored, they are initially saved in the staging area until they are migrated to the LBOSDATA area. While the objects are still in the staging area, the Object Server database has a reference to the physical file in the staging area.

If your system has objects which only reside in the staging area, and the objects have not yet been destaged, a full backup of your system must include both the LBOSDATA folders as well as the staging area!

The objects can be moved to the LBOSDATA directories using the `destage` command.

In addition to these new objects which only reside in the staging area, the staging area may also contains items retrieved (or staged) from archived storage, such as TSM. These objects reside in both the staging area and the LBOSDATA area or TSM, and the Object Server tables point to the physical file, not the staging area.

These objects can be removed from the staging area using the `purge` command.

---

**Attention:** The command mentioned above might take a long time to run; therefore, make sure you include time to run it in your migration plan. The amount of time it takes to perform the `chown` command is dependent on various factors such as number of objects, disk subsystem, amount of memory, and the number of inodes cached. It is difficult to predict the time it takes.

The object files are accessible only by Content Manager Version 8.3. If the migration fails, and you need to go back to Content Manager Version 7.1, you need to run the `chown` command again to change the owner back to osadmin so that the files can be accessed by Content Manager Version 7.1.
Running the destage command
The frequency for destaging objects in a Content Manager system is set in the Content Manager System Administration Client. In this case, we want to have the destage process to start right away. This can be done via a command in the Content Manager Command Utility, as follows:

1. Start the Content Manager Command Utility:
   
   **Start → Programs → IBM Content Manager for Multiplatforms → Command Utility**
   
   The Command Utility windows appears (see Figure 14-3).

   ![Figure 14-3 Destaging objects from the Content Manager Command Utility](image)

2. Connect to the Object Server:
   
   `connect <objsrvrn>` or `c <objsrvrn>`

   Where `objsrvrn` is the name of your Object Server.

3. Start the destaging process:
   
   `destager start`

   The destaging process now copies the eligible files (new objects) from the staging area to the LBOSDATA directory.

   The results of running the destaging process are that:
   
   - The new objects are copied from the staging area to the LBOSDATA directory.
   - The read-only attribute is removed from the files in the staging area once they have been copied to the LBOSDATA directory.
At this point, there should be no files in the staging area with the read-only attribute.

**Running purger command**

In order to remove the files from the staging area, we use the *purger* command. This command relies on a number of settings for the staging area which are set in the Content Manager System Administration Client.

The window shown in Figure 14-4 displays a list of values controlling when the purge process removes files from the staging area. The percentages shown in the frame Purge rate controls when a purge starts. In the example shown, the system starts deleting files when the size of the staging area is 80% or more of the maximum size, 199 MB, which is approximately 160 MB. The system stops deleting files when the size of the staging area is 60% or less of the maximum 199 MB is used, which is approximately 120 MB.

![Figure 14-4 Settings that control the staging area purge process](image)

If you used these settings, and issue the *purge* command while the staging area is less than 80% full, nothing happens. Even if the staging area is more than 80% full, the system stops deleting files when the staging area gets down to 120 MB in size. In order for our manually started purging process to clean the entire staging area, we must reset the purge rate to the values shown in Figure 14-5.
To update the settings for staging area, do the following steps:

1. Stop the Object Server service.
   
   You must close the Command Utility if it is active from the previous step in order to make the Object Server stop.

2. Start the SMS server service.

3. Start the Content Manager System Administration Client and change the values as seen in Figure 14-5 and click Apply.
   
   This forces the purger process to keep deleting files until there are no more files remaining in the staging area.

4. Stop the SMS server service.

5. Restart the Object Server service.

You can now start the purger from the Content Manager Command Utility window:

1. Start the Content Manager Command Utility console:

   **Start → Programs → IBM Content Manager for Multiplatforms → Command Utility**
   
   The Command Utility console appears as shown in Figure 14-6.
2. Connect to the Object Server:
   
   ```
   connect <objsrvrn> or c <objsrvrn>
   ```
   
   Where `objsrvrn` is the name of your Object Server.

3. Start the destaging process:
   
   ```
   purger start
   ```

   ![Figure 14-6 Purging objects from the Content Manager Command Utility](image)

   Once the LBOSDATA directory is empty, the purging process is complete.

   Restore the original values for the staging area using the Content Manager System Administration Client.

### 14.5.3 Shutting down servers

In order to ensure proper migration, you need to shut down:

- **The Content Manager Version 7.1 system**: The Content Manager Version 7.1 production system consists of a Library Server and one or more Object Servers. Before you start the data migration, you need to ensure that no users and no processes are changing data on the system. To achieve this state, you must shut down the Library Server and all Object Server(s).

- **The Content Manager Version 8.3 Resource Manager**: The Resource Manager is a Web application that runs on a WebSphere Application Server. Before you start the migration of the Object Server database to the Resource Manager database, you should stop the application server that is running the Resource Manager Web application.
14.6 Migrating from one Windows machine to another

This section deals with migrating Content Manager Version 7.1 on Windows from one physical machine to Content Manager Version 8.3 on a different physical machine still running Windows. This includes moving both the Library Server and the Object Server to different physical machines. Along the way, we point out any differences in performing this type of migration and performing a standard Content Manager Version 7.1 to Version 8.3 migration, where the Object Server remains on the same machine as the new Resource Manager.

When you migrate from Content Manager Version 7.1 on Windows to Version 8.3 on a different physical machine, you must make sure the prerequisite software levels on the Version 7.1 machine are at the required level.

If you are moving your Content Manager Version 7.1 server (both the Library Server and Object Servers) to a different physical machine, the only software product you may have to upgrade on the Version 7.1 system, is DB2. On the Version 7.1 server, DB2 needs to be at V7.2 with Fix Pack 10 installed (or higher). For instructions on how to upgrade DB2, consult your DB2 documentation. For instructions on how to test for which level of DB2 you are currently running, see A.1.1, “Content Manager prerequisites for Windows” on page 602.

To prepare the target Version 8.3 server, whether this is the same or different physical machine to your existing Version 7.1 Content Manager system, you need to complete a Version 8.3 Content Manager installation on that server.

**Important:** Remember that a Content Manager migration requires an existing Content Manager Version 7.1 system and a fully functioning Version 8.3 system. The migration tools only copy the system and user defined data from one system to another. Both systems must be in good working order before performing a migration.

Because the migration utilities cannot merge data from one system into an existing Resource Manager system, if you already have data in the target system, the utilities overwrite the existing data.

In our migration example, both the Content Manager Library Server and Object Server are on the same physical machine. On our target Version 8.3 server, we install both the Version 8.3 Library Server and Resource Manager components. If you currently have a distributed Version 7.1 system, you need to install only the Content Manager Version 8.3 components that you require.
For example, if you currently have two Object Servers that you wish to move to different physical machines, you only need to install the Resource Manager component on the target machines. Remember, Content Manager prerequisites are needed for a Version 8.3 Resource Manager to function correctly.

If you are not moving either your Library Server or Object Servers to different physical machines, you need to upgrade the prerequisite software on these servers to a level supported by Content Manager Version 8.3. Remember to only upgrade Content Manager to Version 7.1 Fix Pack 23 and DB2 to Version 8.1 Fix Pack 7A. Perform a Content Manager Version 8.3 installation.

Whether you perform a migration that moves your existing Content Manager servers to different physical machines or not, it is important to test that the Version 8.3 Content Manager server you are migrating to is functioning correctly. To do this, test importing a document into the Version 8.3 system and creating an item type. If you are able to do this without any errors, you can consider the Version 8.3 installation a success. You can also use the Content Manager's First Steps utility to import test data into your Version 8.3 system to validate the system. When the migration is performed, all user definitions within the Version 8.3 are overwritten. Do not leave data in the Version 8.3 server that you wish to retain.

Re-visit 14.5, “General data migration preparation” on page 361. Make sure you completed all the steps outlined in that section. If you installed new software versions or Fix Packs on existing or target systems, we recommend that you perform an incremental backup. Make sure no users are logged into both the new and the old systems.

### 14.6.1 Establishing a connection to Version 8.3 Library Server database

Before executing the migration utility, you must complete the following steps:

1. Edit the cmbicmsrvs.ini file, which is extracted from the migration package. The migration wizard uses information from each file to connect to the Content Manager Version 8 system. The next section provides examples of each file.

   **Example 1. cmbicmsrvs.ini file**
   ```text
   ICMSERVER=ls154
   ICMSERVERREPTYPE=DB2
   ICMSCHEMA=ICMADMIN
   ICMSO=FALSE
   ICMBAUTH=SERVER
   ICMBREMOTE=FALSE
   ICMBHOSTNAME=
   ICMBPORT=
   ```
ICMREMTEDB=
ICMNODENAME=
ICMOSTYPE=

Now, you need to establish a connection to the Version 8.3 Library Server database on the Version 7.1 system. This is because the first migration utility runs on the Version 7.1 system and must have a connection to both databases when it runs.

**Important:** This step is only applicable if you migrate a Version 8.3 Library Server running on a different machine. It is *not* required if you have installed your Content Manager Version 8.3 Library Server on the same machine as your Version 7.1 Library Server.

Verify that the database server is running. Establish a DB2 connection as follows:

1. Start the DB2 Client Configuration Assistant:
   
   Select **Start → Programs → IBM DB2 → Client Configuration Assistant**

   The DB2 Client Configuration Assistant window opens (see Figure 14-7). This window gives you a list of existing databases that have been defined to the system.

![Figure 14-7 DB2 Client Configuration Assistant](image-url)
2. Select **Add** to define a new database entry.

The Add Database Wizard starts and opens the Source panel (see Figure 14-8).

![DB2 Add Database Wizard: Source](image)

**Figure 14-8**  DB2 Add Database Wizard: Source

3. Select **Manually configure a connection to a database**, and then click **Next** to continue.

The Protocol panel opens (see Figure 14-9).
4. Select **TCP/IP** as the communication protocol that we use to connect to the Content Manager Version 8.3 Library Server database. Click **Next**.

The TCP/IP panel opens (see Figure 14-10).
5. Enter the fully qualified host name of the Content Manager Version 8.3 Library Server, and the port the DB2 instance is listening on. The default port for DB2 is 50000. Click Next.

The Database panel opens (see Figure 14-11).
6. Enter the Content Manager Version 8.3 database name and alias. By default in DB2, the alias of a database is the same as its name. Click Next.

The ODBC panel opens (see Figure 14-12).
Figure 14-12  DB2 Add Database Wizard: ODBC

7. **Keep the default values for the ODBC definitions. Click Next.**

The Node Options panel opens (see Figure 14-13).
8. The node information is optional. In our scenario, we leave the fields blank. Click Next.

The Security Options panel opens (see Figure 14-14).
9. Select **Configure security options (Optional)** and then select **Use authentication value specified in the server's DBM configuration**. Click **Finish**.

A confirmation message appears informing you that your configuration for the particular database you specified is added successfully (see Figure 14-15).

10. Now that you have successfully added the definition of your database, you should test the connection to it. Click **Test Connection**.
11. Enter the User ID and Password in the Connect window (see Figure 14-16). Click OK.

![Connect To DB2 Database](image)

*Figure 14-16  DB2 Add Database Wizard: Test connection to database*

You should see a message appear, stating that the connection test is successful (see Figure 14-17).

![DB2 Message](image)

*Figure 14-17  DB2 Add Database Wizard: Connection to database successful message*

12. Click OK.
14.6.2 Running the Migration Wizard

Once you have prepared for the migration and established a DB2 connection, it is time to run the first migration utility, the Migration Wizard.

The Migration Wizard assists you through the following steps:

- “Step 1: Preparing for migration” on page 382
- “Step 2: Testing communication and verifying authorization” on page 383
- “Step 3: Generating a report of non-migratable tables” on page 385
- “Step 4: Identifying storage location for migration files” on page 387
- “Step 5: Setup for migrating system setup tables” on page 388
- “Step 6: Setup for migrating system setup tables continued” on page 389
- “Step 7: Migrating Object Servers to Resource Managers” on page 391
- “Step 8: Migrating system definition data” on page 392
- “Step 9: Migrating data tables” on page 394
- “Step 10: Instructions for completing the migration process” on page 395

**Important:** The migration program uses the user ID of the current user when accessing the database. You must log on to the system using the user ID you used when creating the Content Manager Version 7.1 Library Server database, because this user ID is reflected in the schema of the Content Manager Version 7.1 Library Server table definitions.

You must run the Migration Wizard from the machine where the Content Manager Version 7.1 Library Server is installed.

**Launching the Migration Wizard**

Launch the Migration Wizard from your Version 7.1 system:

1. Open a command window.
2. Change to the migrate directory.

   This is the directory where you unzip or uncompress the file downloaded from the URL mentioned at 14.3.1, “What is new in the migration tool for Content Manager V8.3” on page 357.
3. Enter the command:

```
frn2icml
```

The Migration Wizard starts and the Preparation panel opens (see Figure 14-18).

---

**Step 1: Preparing for migration**

Proceed as follows:

1. Read all of the instructions on this panel carefully. Pay special attention to the list of three actions that must be completed before progressing with the migration.

2. Click **Next** to go to the next step.

   The communications panel opens (see Figure 14-19).

---

**Note:** Use the **Back** button (where available) in any of the displayed panels throughout the wizard in order to return to previous screens.
Step 2 of 10: Communication and authorization verification

This wizard must connect to both the library server database and library server of both the earlier Content Manager Version and Content Manager Version 8.3. You must provide an Administrator’s user name that has super user access privileges. Click Verify. After authorization for both systems has been verified, click Next to continue.

1. For Content Manager Version 6.1 or Version 7.1, enter the following information:
   - Library Server database name
   - Database administrator user name
   - Database administrator password

2. For Content Manager Version 8.3, enter the following information:
   - Library Server name
   - Administrator user name
   - Administrator password
   - DB2 UDB schema (the default for this is ICMADMIN)

   Important: The database schema must be entered in upper case.

3. Click Verify to start the verification process. This process verifies the connection to both the Version 7.1 (or Version 6.1) Library Server database and the Version 8.3 Library Server database, using the supplied user IDs and passwords.

Figure 14-19 Content Manager migration step 2 (frn2icml): Communications

Step 2: Testing communication and verifying authorization

Proceed as follows:

1. For Content Manager Version 6.1 or Version 7.1, enter the following information:
   - Library Server database name
   - Database administrator user name
   - Database administrator password

2. For Content Manager Version 8.3, enter the following information:
   - Library Server name
   - Administrator user name
   - Administrator password
   - DB2 UDB schema (the default for this is ICMADMIN)

   Important: The database schema must be entered in upper case.

3. Click Verify to start the verification process. This process verifies the connection to both the Version 7.1 (or Version 6.1) Library Server database and the Version 8.3 Library Server database, using the supplied user IDs and passwords.
4. Click **Next** when the Verification status on both connections changes to **Authorized**.

**Attention:** If communication fails with the earlier Content Manager Library Server, you can view the migrate.err file or errors, which is created within the migrate directory from which you ran the `frn2icml` command.

If communication fails with the Content Manager Version 8.3 Library Server, the Verification status field displays an SQL error message. For more information about the message, refer to the following documentation:

*IBM DB2 Universal Database - Message Reference Volume 1, GC09-2978*
*IBM DB2 Universal Database - Message Reference Volume 2, GC09-2979*

If authorization fails, verify that the database user name that you entered exists, has administrative privileges, and that the password you entered is correct.

The non-migratable tables panel opens (see *Figure 14-20*).

![Figure 14-20](content-manager-migration-step-3-frn2icml-non-migratable-tables.png)

*Figure 14-20  Content Manager migration step 3 (frn2icml): Non-migratable tables*
Step 3: Generating a report of non-migratable tables

Proceed as follows:

1. Click **Generate Report** to view a list of database tables that will not be migrated (see Figure 14-21). It is unlikely that any of these tables contain data relevant for the new version of Content Manager. The use of these tables in the earlier Content Manager versions would have required custom application programs using API calls. If you are unsure whether or not you require the data in these tables, back them up.

![Non-migratable Tables](image)

*Figure 14-21  Content Manager migration step 3 (frn2icml): Non-migratable tables listing*

During this step, the wizard may detect existing migration data (for example, if you have previously run this wizard) and prompt you to decide what to do with that data (see Figure 14-22). This figure shows a report of non-migratable tables, if you have previously run the frn2icml migration tool.
Whenever this occurred in our scenario (we ran this utility a number of times for testing purposes), we always chose to delete the old data and create new migration data. This had no adverse effects on our Content Manager Version 7.1 system or in the remainder of the migration steps.

If the wizard has trouble during the detection of existing data, it prompts you to select Refresh so that it can try again. The migration utility only deletes the migration related data from the Content Manager Version 7.1 (or Version 6.1) Library Server database. If the wizard does not detect existing migration data, you do not have to make this decision.

**Important:** If you choose to delete this data, make sure that you also delete any migrated data from your Content Manager Version 8.3 system.

2. Click **OK** to close the non-migratable table listing window.
3. Click **Next** to go to the next step.

The storage panel opens (see Figure 14-23).
Figure 14-23  Content Manager migration step 4 (frn2icml): Storage

Step 4: Identifying storage location for migration files

Proceed as follows:

1. Wait while the Migration Wizard calculates the amount of space required for the migration files. This process may take several minutes for very large databases.

2. Once the wizard has finished the estimation, you see the results in the Space required (estimate) field. (Note, in our scenario, the database is extremely small. The value shown above is not realistic for a production database). In the field Save migration files in this location, specify a folder located on a drive that has enough free space available to store the migration files. You can also create a new folder now and use Browse to locate this folder.

3. Read the Attention section carefully. It states that a data file will be created for each Object Server in your Content Manager Version 7.1 (or Version 6.1) system. It also states a very important fact: for migration to succeed, your Content Manager Version 8.3 system must have at least as many Resource Managers as there are objects servers in your Content Manager Version 7.1 (or Version 6.1) system.

Note that the wizard creates a folder for each Object Server and places the Object Server data file within the relevant folder. The folder name is based on
the name of the Object Server. It does not actually create the folders until later on in the wizard.

4. Once you have carefully read this panel, and are happy that the location you have specified for the migration files is of adequate capacity, click **Next** to go to the next step.

The system setup table panel opens (see Figure 14-24).

---

![Figure 14-24](Content Manager migration step 5 (frn2icml): System setup tables)

**Step 5: Setup for migrating system setup tables**

Proceed as follows:

1. In the **Client Code page** field, enter the code page used by your Version 7.1 (or Version 6.1) Content Manager clients. In our scenario, it is 1252.

   Selecting the correct code page ensures the proper display of text notes on your Content Manager Version 8.3 clients.

2. In the **Language code** field, enter the primary language in which the names for the data modeling objects are defined in your earlier Content Manager System Administration Client. This language code is used during the creation of your data model, so names and labels are written in this language.

   Selecting the correct language code ensures proper display of the data model names and labels.
3. Once you have completed both fields, click **Next** to go to the next step.

The system setup tables 2 panel opens (see Figure 14-25).

![Content Manager Version 8 Database Migration Wizard](image)

**Figure 14-25  Content Manager migration step 6 (fn2icml): System setup tables 2**

**Step 6: Setup for migrating system setup tables continued**

Proceed as follows:

1. If you use the item names capability of Content Manager Version 7.1, you need to determine whether you want to migrate the item names or not.

   Consider the following factors before doing so:
   
   - Content Manager Version 8.3 does not include the item name capability; therefore if you select this check box, the item names are migrated as item attributes.
   
   - Items in Content Manager do not contain the item name as a system defined attribute. If you choose to migrate item names, the Migration Wizard defines an item name as a user-defined attribute in the root component of all item types. The item name value from Content Manager Version 7.1 is placed in this attribute. So if you performed a search using the Content Manager Version 8.3 Windows client after the migration, you would see an extra attribute displayed in the search results list.

   Depending on the factors mentioned above, decide whether or not to select the Migrate item names check box.
2. In the Grant privilege set drop-down list, select a default grant privilege set for the users that you are migrating. A grant privilege set specifies the privileges that users can grant to users that they create. Grant privilege sets is a new feature in Content Manager Version 8.3.

3. Once you have decided whether or not to enable the migration of item names, and have selected the default grant privilege set for migrated users, click **Next** to go to the next step.

The map servers panel opens (see Figure 14-26).

---

**Figure 14-26  Content Manager migration step 7 (frn2icml): Map servers**
Step 7: Migrating Object Servers to Resource Managers

Proceed as follows:

1. In this step you need to map each earlier Content Manager Object Server with a Content Manager Version 8.3 Resource Manager. To map an Object Server with a Resource Manager, perform the following steps:
   a. Select an Object Server from the Object servers list.
   b. Select a Resource Manager from the Resource Managers list.
   c. Click Map.

   If you attempt to map an Object Server with a Resource Manager that has a different host name, the wizard prompts you for verification before proceeding (see Figure 14-27).

   ![Figure 14-27](Warning.png)

   Figure 14-27  Migration step 7 (frn2icml): Warning during mapping OS with RM

   Since we want to move our Object Server to a different physical machine, we receive the warning above. Click Yes if you are attempting this type of migration. If you are performing an officially supported Object Server migration, where the Object Server and Resource Manager reside on the same physical machine, you should not receive this warning message.

   You must have at least as many Resource Managers as Object Servers. If you do not, the wizard will inform you to add a Resource Manager or remove an Object Server and select Refresh.

2. Click Next when you have mapped all of your existing Object Servers to Resource Managers.

   The system definition data panel opens (see Figure 14-28).
Figure 14-28  Content Manager migration step 8 (frn2icml): System definition data

**Step 8: Migrating system definition data**

Proceed as follows:

1. Click **Migrate System Table** to migrate your earlier Content Manager system definition data.

   The Migration Wizard uses the Content Manager Version 8.3 stored procedures to create the Version 8.3 entities. See the Content Manager Version 8.3 Library Server log file (ICMSERVER.LOG - the default location for this file is C:) to see what errors occurred during this step of the migration process. For more information about errors logged in the log file, see Messages and Codes. Figure 14-28 shows this panel after the step is completed successfully.
2. **Click Next** when all of the system definition data has been successfully migrated.

The data tables panel opens (see Figure 14-29).

**Attention:** If you have previously run the frn2icml tool against a Content Manager Version 8.3 Library Server database, and are attempting to run the tool again against the same database, it is possible that you could receive an error when migrating the system tables again.

If you experience this problem, the easiest way to solve it is to reinstall the Content Manager Version 8.3 Library Server database. This gives you a clean database to perform this step.

**Important:** The Version 8.3 Library Server database should not hold any production data prior to a migration; or else you will lose the data when you reinstall the database.
**Step 9: Migrating data tables**

Proceed as follows:

1. This step prepares the user data, from the previous version of your Content Manager Library Server, for migration to your Content Manager Version 8.3 Library Server. The table within the panel gives you an estimate of how long this process takes, and the amount of disk space that it requires.

   **Important:** All data migration must be completed in a single migration session; therefore only prepare the data tables if the required time and disk space exists for completing the entire data migration. For example, if your earlier Content Manager server needs to be back in production shortly, there may not be enough time to complete this migration step before the server must be available for production again. In this case, you may want to wait until a later time.

   Once you are satisfied that you can accommodate the space and time requirements, click **Prepare Data Tables**.

2. At the end of this step, all of the Library Server files are placed in a single Library Server directory which is named using your current Version 7.1 or Version 6.1 Library Server name. All of the Object Server files, one for each Object Server, are placed in a single Object Server directory which is named using your current Version 7.1 or Version 6.1 Object Server name(s). There is a separate directory for each Object Server if there are multiple Object Servers.

   Click **Next** to go to the next step.

   The completion panel opens (see Figure 14-30).
Figure 14-30  Content Manager migration step 10 (frn2icml): Completion

**Step 10: Instructions for completing the migration process**

Proceed as follows:

1. Read this panel carefully. Click **Print Instructions** to print out the steps that you should follow to import the migrated data from the identified directory into Content Manager Version 8.3.

2. Click **Exit** to close the wizard.

### 14.6.3 Importing user data into new Library Server

The first migration utility (frn2icml) migrates the Library Server system definitions from your earlier version of Content Manager to Content Manager Version 8.3 and exports user data to DEL files. If you log into a Content Manager Version 8.3 System Administration Client, you should see your system definitions, such as user IDs, item types, and attributes. Now we need to load our user data (or metadata) into Content Manager Version 8.3.

In this section, we discuss how to import our user data into the new Content Manager Version 8.3 Library Server.
This section is divided into three sub-sections:

- “Summary of files generated by the Migration Wizard” on page 396
- “Transferring files created by the Migration Wizard” on page 399
- “Running the Library Server import utility” on page 400

**Summary of files generated by the Migration Wizard**

There are a number of data files generated by the Migration Wizard. Some are for Library Server, some are for Object Server(s).

Figure 14-31 lists the files generated by the Migration Wizard in our scenario for Library Server. These are the source files used to import user data into Content Manager Version 8.3.

![Figure 14-31](image)

**Figure 14-31   Contents of Library Server folder generated by Migration Wizard (fm2icml)**

As seen from Figure 14-31, the Migration Wizard creates a folder named libsrvrn. This is the name of our Content Manager Version 7.1 Library Server. The Migration Wizard places the folder in the location that we specify in step 4 of the Migration Wizard (see “Step 4: Identifying storage location for migration files” on page 387 as reference).
Later in the migration process, we need to copy these files to our target Library Server in order to load data into the Content Manager Version 8.3 Library Server. You only need to do this if you plan to move your Library Server to a different machine.

The DEL files are used as the import source for the next stage of the migration; the ERR files contain messages about the number of rows exported. We recommend briefly viewing the contents of these ERR files and checking for any errors.

The number of files created in the Library Server folder is dependent on the number of index classes you have. In our Version 7.1 system, we only created one index class in addition to the system defined index classes. In your situation, you most likely have more files than we have in Figure 14-31.

The Migration Wizard log file (migration.log) should also be browsed for any errors. The migration utility writes this log file to the location from which you run frn2icml. In our scenario, the directory is F:\Migrate\DB2 (see Figure 14-32).

As you can see from the modified value of the files, three new files are created by the Migration Wizard:

- cm7bind.err
- cm8bind.err
- migration.log
In addition to reviewing the migration.log file, we also recommend reviewing the *bind.err files listed above. These files inform you if any errors occurred while the wizard perform the necessary bind operations on the databases.

**Attention:** As you can see from Figure 14-31 on page 396, the Migration Wizard no longer combines the DEL files that it creates into a JAR file as earlier versions of this wizard used to; instead, folders are created using the names of the Library Server and Object Server(s) and the DEL files are placed into these folders.

Figure 14-33 lists the file generated by the Migration Wizard in our scenario for the Object Server. This is the source file for importing data into Content Manager Version 8.3.

![Figure 14-33 Contents of Object Server folder generated by Migration Wizard (frn2icml)](image)

As shown in Figure 14-33, the Migration Wizard creates a folder using the name of the Object Server database that we define in our system (objsrvrn). The wizard places this folder in the location we specified in step 4 of the Migration Wizard (see “Step 4: Identifying storage location for migration files” on page 387 for reference).

If we have more than one Object Server defined within our earlier Content Manager system, the Migration Wizard would have created a folder and a DEL file for each of the respective Object Servers.

Later in the migration, we need to copy this file to our target Resource Manager in order to load data into our Content Manager Version 8.3 Resource Manager. Note that you only need to do this if you plan to move your Resource Manager to a different machine.
Transferring files created by the Migration Wizard

In our scenario, we are moving the Version 7.1 Content Manager Library Server to a different machine during the migration process. We need to copy the files that the Migration Wizard created on the Version 7.1 system, to our Content Manager Version 8.3 Library Server machine. The files that the Migration Wizard created, in our scenario, are located at F:\WinMig\libsrvrn.

**Attention:** You must run the Library Server import utility (icmimp1) from the Content Manager Version 8.3 Library Server machine. If you are not moving your earlier Content Manager Library Server to a different machine, the Library Server import utility runs from the same machine as the Migration Wizard (frn2icml).

If you are *not* planning to move your earlier Content Manager Library Server to a different machine, copy the contents of your Library Server folder created by the wizard to your migrate directory to prepare to run the Library Server import utility.

Perform the following steps when planning to move to a different Library Server machine as part of your Content Manager migration process:

1. Copy the migrate folder to any location on your Content Manager Version 8.3 Library Server. In our scenario, we copy the files to G:\MIGRATE directory.

2. Copy the Library Server export files created by the Migration Wizard (frn2icml), located in our scenario, F:\WinMig\libsrvrn, to the DB2 folder within the migrate folder that you copied to your Version 8.3 Content Manager Library Server in step 1. In our scenario, we copy the files to G:\MIGRATE\DB2 directory. Only the DEL files are needed by the import utility; however, it is easier to copy over the entire libsrvrn folder.

See Figure 14-34 for the contents of our DB2 folder on our Content Manager Version 8.3 Library Server after completing this step.
Now that we have transferred the files to our target Content Manager Version 8.3 Library Server, we are in a position to run the Library Server import utility as follows:

1. Make sure that DB2 is started on your Content Manager Version 8.3 Library Server machine, before completing these steps.
2. Open a DB2 Command Window:

   **Start → Programs → IBM DB2 → Command Line Tools → Command Window**

3. Change the directory to the location of your DB2 folder that you copied to your Content Manager Version 8.3 Library Server earlier. In our scenario, it is G:\MIGRATE\DB2.

4. Enter the command:

   ```
   icmimpl CM8LSNAME CM8DBADMINID CM8ADMINPW SCHEMANAME
   ```

   Where

   - **CM8LSNAME** is the name of the Content Manager Version 8.3 Library Server database.
   - **CM8DBADMINID** is the database administrator user ID used to create the Content Manager Version 8.3 Library Server database tables.
   - **CM8ADMINPW** is the password of the database administrator user ID used to create the Content Manager Version 8.3 Library Server database tables.
   - **SCHEMANAME** is the name of the database schema used to create the Content Manager Version 8.3 Library Server database tables.

   The import process now imports user data into the Content Manager Version 8.3 Library Server tables. You will see a number of command windows briefly appear. Example 14-1 shows the output of the `icmimpl` command.

**Example 14-1 Our output from the ICMIMPL command**

```
F:\MIGRATE\DB2>icmimpl cmldb05 icmadmin icmadmin ICMADMIN
Loading data from ICMSTCHECKEDOUT.DEL into ICMSTCHECKEDOUT
Loading data from ICMSTITEMS12.DEL into ICMSTITEMS001001
Loading data from ICMSTITEMS13.DEL into ICMSTITEMS001001
Loading data from ICMSTITEMS6.DEL into ICMSTITEMS001001
Loading data from ICMSTITEMS8.DEL into ICMSTITEMS001001
Loading data from ICMSTITEMSPARTS.DEL into ICMSTITEMS001001
Loading data from ICMSTLINKS001001.DEL into ICMSTLINKS001001
Loading data from ICMSTRI001001.DEL into ICMSTRI001001
Loading data from ICMUT00204001.DEL into ICMUT00204001
Loading data from ICMUT00300001.DEL into ICMUT00300001
Loading data from ICMUT01000001.DEL into ICMUT01000001
Loading data from ICMUT01005001.DEL into ICMUT01005001
Loading data from ICMUT01007001.DEL into ICMUT01007001
Loading data from ICMUT01009001.DEL into ICMUT01009001
Loading data from ICMUT01010001.DEL into ICMUT01010001
F:\MIGRATE\DB2>
```
The import process generates the following output:

- Log files, one for each of the DB2 imports performed. The log file is named 
  <the original file name minus the DEL extension>LOAD.ERR.
- A general log file called migration.log.
- A database bind log named cm8bind.err.

All of these files listed are created in the directory in which you run the import 
process (icmimpl). In our scenario, they are created in F:\Migrate\DB2 as shown 
in Figure 14-35.

![Figure 14-35](image)

We recommend reviewing these log files and making sure there are no error 
messages.

The migration of the Library Server from Content Manager Version 7.1 (or 
Version 6.1) to Content Manager Version 8.3 is now complete. You can now log 
into a Content Manager Version 8.3 Windows client and search through your 
meta data inside your item types to display search results lists.

**Important:** Remember that you cannot display any objects yet, as we have 
not performed the Objects Server migration step yet. Do not open any 
documents!
14.6.4 Migrating user data into new Resource Manager(s)

In this section we discuss how to migrate a Content Manager Version 7.1 (or Version 6.1) Object Server to a Content Manager Version 8.3 Resource Manager. During this process, the migration process removes any existing data you may have in your Content Manager Version 8.3 Resource Manager tables. For this reason, you should not store any production data in the new Content Manager Version 8.3 Resource Manager before running the migration.

This section is divided into three sub-sections:

- “Migrating to a Resource Manager on the same machine” on page 403
- “Migrating to a Resource Manager on a different machine” on page 407
- “Copying objects” on page 410 (applicable only if migrating to a Resource Manager on a different machine)

The first section describes how to perform a supported Object Server migration, where the existing Object Server(s) and the new Version 8.3 Resource Manager(s) exist on the same machine(s). In this instance, the standard Resource Manager migration utility can be used (icmimpo), and the physical objects stored within the LBOSDATA directory remain in the same location.

The second section describes how to migrate an Object Server to a different physical machine. In this instance, we need to run the (icmimpo) utility in two separate ways:

1. Run a command (icmmidr) on the original Object Server machine.
2. Copy the output from this machine to the new Resource Manager.
3. Run the command (icmmidr) again, but on the new Resource Manager.

In our scenario, we have one Library Server and one Object Server installed on the same machine. If you have a distributed system with more than one Object Server, you need to perform these steps once for each Object Server.

The third section describes how to copy objects from the existing system to the new system.

**Migrating to a Resource Manager on the same machine**

Use the steps below in order to migrate a Content Manager Version 7.1 (or Version 6.1) Object Server to a Content Manager Version 8.3 Resource Manager.
If you have more than one Object Server for your system, you need to perform these steps once on each Object Server:

1. Copy the Object Server output generated by the Migration Wizard from the earlier Content Manager Library Server to the DB2 folder within the migrate folder.

   If you currently have your Library Server and Object Server on the same machine, then you have already copied the migrate directory to your machine. If you are performing this step on a machine with just an Object Server installed, you first need to copy this migrate directory to your machine and then copy over the relevant Object Server file (you can copy it to any location). If you have more than one Object Server, you need to make sure you copy over the correct Object Server file created by the Migration Wizard. Remember the Migration Wizard creates a folder for each Object Server you have and the folder’s name is based on the name of your existing Object Servers.

2. During the Object Server database migration, the migration utility creates a table in your Content Manager Version 7.1 (or Version 6.1) Object Server database and loads the data into this table. An error during the load process may put the table space in a locked state and may deny access to other tables in that table space. For this reason, we recommend that you create this new table in a separate table space.

   This means creating a DB2 table space in each Object Server databases that you have. The table to be created within this table space has approximately the same size as your existing Object Server database. Make sure you have adequate disk space on each of your Objects Servers to accommodate this. When you run the Object Server migration utility, you specify the name of the table space you have created as one of the parameters. To create a table space, use the following steps:

   a. Create a folder to be used as a storage area for the table space. For performance reasons, we recommend that you create this folder on a different volume than the current Content Manager Version 7.1 (or Version 6.1) Object Server database. Remember that the drive on which you decide to create the table space must have enough space to accommodate the size of the current Object Server database.

   b. Start the DB2 Control Centre:

      Start → Programs → IBM DB2 → Control Centre

   c. Expand the tree in the left hand window until you reach your Object Server database and then expand the Object Server database. Right-click Table Spaces, and then select Create → Table Space Using Wizard (see Figure 14-36).
d. Create the table space using the wizard. Specify the folder you created in step one as the location of the table space. For other values, we use default in our scenario.

3. Open a DB2 Command Window:
   
   Start $\rightarrow$ Programs $\rightarrow$ IBM DB2 $\rightarrow$ Command Window

4. Change the directory to your DB2 directory. This is the directory in which you copied your Object Server file generated by the Migration Wizard earlier.

5. Enter the command:
   
   icmimpo CM7OSNAME CM7OSADMINID CM7OSADMINPW CM7TBLSPACE CM8RMNAME CM8RMADMINID CM8RMADMINPW

   Where

   CM7OSNAME is the name of the Content Manager Version 7.1 (or Version 6.1) Object Server.

   CM7OSADMINID is the database administrator user ID used to create the Content Manager Version 7.1 (or Version 6.1) Object Server database tables.
CM7OSADMINPW is the password of the database administrator user ID used to create the Content Manager Version 7.1 (or Version 6.1) Object Server database tables.

CM7TBLSPACE is the table space where the migration related tabled should be located. This is the name of the table space you have just created on your Content Manager Object Server.

CM8RMNAME is the name of the Content Manager Version 8.3 Resource Manager.

CM8RMADMINID is the database administrator user ID used to create the Content Manager Version 8.3 Resource Manager database tables.

CMRMADMINPW is the password of the database administrator user ID used to create the Content Manager Version 8.3 Resource Manager database tables.

The migration process now migrates user data from the Content Manager Version 7.1 (or Version 6.1) Object Server database into the Content Manager Version 8.3 Resource Manager tables. You will see a number of command windows briefly appear. Example 14-2 shows the output of the icmimpo command when we run it against our Version 8.3 Resource Manager database.

Example 14-2  Our output from the ICMIMPO command

F:\Migrate\DB2>icmimpoobjsrvr db2admin db2admin3 MIGTS cmrmdb05 radmin rmadmin
WARNING: The CM Version 8.3 Resource Manager only supports 'Password Prompt' mode for TSM access. If you are using TSM using TSM is 'Password Generate' mode, you must switch to 'Password Prompt' mode before proceeding with migration
Press <Q> to quit the migration utility to verify your TSM mode or to change to 'Password Prompt' mode.
Type a letter and ENTER to continue with the migration process
f
Continuing with Migration
Creating CM parts mapping table
Loading CM parts mapping table
Exporting data from CM7 Object Server database
Loading data in the CM8 Resource Mgr database

F:\Migrate\DB2>

This completes the Content Manager Version 7.1 (or Version 6.1) Object Server migration. We recommend reviewing the log files for error messages. These log files are created by the icmimpo command within the same directory from which you run it from. In our scenario, the log files are in F:\Migrate\DB2 (see Figure 14-37).
Chapter 14. Upgrade and migration on multiplatforms

Figure 14-37 Log files created by the Resource Manager import utility (icmimpo)

**Migrating to a Resource Manager on a different machine**

This approach is not officially supported; however, we want to document it here because we believe this situation occurs quite often in real world environment.

Because the Object Server migration utility (icmimpo) must be run on the same physical machine as the current Content Manager Version 7.1 (or Version 6.1) Object Server and the Content Manager Version 8.3 Resource Manager, we must run the command in a slightly different way when the Object Server and Resource Manager are on different machines. Simply creating a database connection to a remote Resource Manager and running this utility does not work.

The Object Server migration utility first exports data from the current Object Server database and then imports it into the Resource Manager database. The icmimpo command runs a batch file (icmimpo.bat) which in turn calls an executable icmnmidr twice. You can see this by editing the icmimpo.bat file (see Figure 14-38).
As you can see the `icmnmidr` executable is called twice within this batch file, once to export the Object Server data and once to import this exported data into the new Resource Manager.

This gives us the ability to run the first executable on the Content Manager Version 7.1 (or Version 6.1) Object Server, copy the files created by this executable over to our target Content Manager Version 8.3 Resource Manager, and then run the executable again to import this data into the Resource Manager database.

To do this:

1. Follow steps 1 to 4 of, “Migrating to a Resource Manager on the same machine” on page 403.
2. Enter the command:
   
   ```
   icmnmidr 0 CM7OSNAME CM7OSADMINID CM7OSADMINPW CM7TBLSPACE
   ```

   Where:

   - **CM7OSNAME** is the name of the Content Manager Version 7.1 (or Version 6.1) Object Server.
   - **CM7OSADMINID** is the database administrator user ID used to create the Content Manager Version 7.1 (or Version 6.1) Object Server database tables.
CM7OSADMINPW is the password of the database administrator user ID used to create the Content Manager Version 7.1 (or Version 6.1) Object Server database tables.

CM7TBLSPACE is the table space where the migration related tabled should be located. This is the name of the table space you have just created on your Content Manager Object Server.

The icmnmidr executable now exports user data from the Content Manager Version 7.1 (or Version 6.1) Object Server database and stores this information within the DEL files that are stored in the same directory from which you run the executable. You will see a number of command windows briefly appear. Example 14-3 shows the output of the icmnmidr command when we run it against our Version 7.1 Object Server database.

Example 14-3  Our output from the icmnmidr executable run on our V7.1 Object Server

F:\Migrate\DB2>icmnmidr 0 objsrvrn db2admin db2admin3 MIGTS WARNING: The CM Version 8.3 Resource Manager only supports 'Password Prompt' mode for TSM access. If you are using TSM using TSM is 'Password Generate' mode, you must switch to 'Password Prompt' mode before proceeding with migration. Press <Q> to quit the migration utility to verify your TSM mode or to change to 'Password Prompt' mode. Type a letter and ENTER to continue with the migration process.
Continuing with Migration
Creating CM parts mapping table
Loading CM parts mapping table
Exporting data from CM7 Object Server database

F:\Migrate\DB2>

3. Copy the files created from running the icmnmidr command to your DB2 directory on your Content Manager Version 8.3 Resource Manager (the DB2 directory from the Content Manager Version 8.3 Windows installation CD).

4. Open a DB2 Command Window and change the directory to this DB2 directory.

5. Enter the command:
   icmnmidr R CM8RMNAME CM8RMADMINID CM8RMADMINPW
   Where:
   CM8RMNAME is the name of the Content Manager Version 8.3 Resource Manager.
   CM8RMADMINID is the database administrator user ID used to create the Content Manager Version 8.3 Resource Manager database tables.
CMRMADMINPW is the password of the database administrator user ID used to create the Content Manager Version 8.3 Resource Manager database tables.

The icmnmidr executable now imports user data from the Content Manager Version 7.1 (or Version 6.1) Object Server DEL files, and stores them into the Content Manager Version 8.3 Resource Manager database tables. You will see a number of command windows briefly appear. Example 14-4 shows the output of the icmnmidr command when we run it against our Version 8.3 Resource Manager database.

Example 14-4

F:\MIGRATE\DB2>icmnmidr R cmrmdb05 rmadmin rmadmin
Loading data in the CM8 Resource Mgr database

F:\MIGRATE\DB2>

This completes the database migration tasks needed in order to migrate a Content Manager Version 7.1 (or Version 6.1) Object Server database to a Content Manager Version 8.3 Resource Manager that resides on a different machine. Now we must copy the objects from the original Object Server over to the Resource Manager.

Copying objects
This step is only applicable if you migrate the Object Server to a Resource Manager on a different machine. After performing the steps described in “Migrating to a Resource Manager on a different machine” on page 407, the local objects from the original Object Server can now be copied over to the Version 8.3 Resource Manager. When you install your new Resource Manager, you specify the location to store the objects (for example C:\icmstorage). As soon as the first object is stored into this new Resource Manager, an LBOSDATA directory is created within the location (in this example, C:\icmstorage).

The supported method of migrating an Object Server does not involve moving objects; therefore, the Object Server migration we go through earlier does not update the LBOSDATA directory. We need to copy our existing LBOSDATA directory and all of its contents from the original location to the location that you specify during the installation of the Version 8.3 Resource Manager for the object storage.

The current location can be anything you specified when you installed the earlier version of the Content Manager. For example, it may be in G:\frndata\storage on the Object Server. Using this example, we copy the LBOSDATA directory from G:\frndata\storage on the current Object Server to C:\icmstorage on the new Version 8.3 Resource Manager. This Content Manager Version 8.3 Resource Manager can now access the objects.
If your LBOSDATA directory is very large, you need to consider which method of copying files is the most efficient way for your particular environment. In our scenario, we use an FTP client. Because we are simply copying files over, this action does not affect the objects stored within the current Object Server; if something does go wrong with the object copy, we can restart the copying process. While you are copying the files, you must make sure that no new files are stored into the existing (current) Content Manager system. If this happens, your new Resource Manager database and the objects within its LBOSDATA directory may not be synchronized.

If you currently use TSM on your Object Server, you need to install the TSM Client APIs on your new Resource Manager server and configure the client options file as it is on your original (current) Object Server. Once this is done, your Version 8.3 Resource Manager can communicate with your TSM server and store and retrieve objects just as your Version 7.1 (or Version 6.1) Object Server did.

There are other methods to move a Resource Manager to a different physical server after the upgrade to Content Manager Version 8.3 is performed. For example, you can create a new Resource Manager on the target machine, then create a storage class on your existing Resource Manager, configure it so that it specifies your new Resource Manager as a remote destination. After doing this, you can update your existing migration policy, specifying this new storage class as the next migration step. You may need to update your existing migration definitions if the retention period is set to forever, as this means the objects never migrate to the next storage class. Specify all new object stores to go to the new Resource Manager, and when all of the existing objects are migrated to the new Resource Manager, you can delete your original Resource Manager.

**Important:** The final part of the migration process is system validation. You must validate your new system to make sure everything is completed successfully.

Refer to 14.9, “Post migration validation” on page 416 for details.

### 14.7 Migrating CM V7.1 on Windows to CM V8.3 on AIX

In this scenario, we migrate a Content Manager Version 7.1 system on a Windows platform, with one Library Server and one Object Server on the same machine, to a Content Manager Version 8.3 system on an AIX platform, with the same configuration. This approach is applicable for a Content Manager Version 6.1 system as well.
While migrating a Content Manager Version 7.1 Library Server on Windows to a
Content Manager Version 8.3 Library Server on AIX is officially supported,
migrating the Object Server from Windows to AIX is not officially supported.
(Note that migrating the Object Server to any other machine, regardless of
operating system, is not officially supported.)

Make sure you read the following sections prior to performing this migration:

1. 14.1, “Introduction” on page 348
2. 14.3, “Migration considerations” on page 357
3. 14.4, “Data migration overview” on page 360
4. 14.5, “General data migration preparation” on page 361
   For the above sections, also follow the general preparation instructions to
prepare for your data migration.

The steps to perform the actual migration are almost identical to the instructions
in 14.6, “Migrating from one Windows machine to another” on page 370.
Specifically, follow the instructions in the following sub-sections:

1. 14.6.1, “Establishing a connection to Version 8.3 Library Server database” on
   page 371
   The only difference is that when you establish a database connection to the
Content Manager Version 8.3 Library Server database, you need to establish
the connection to an AIX DB2 database.

2. 14.6.2, “Running the Migration Wizard” on page 381
3. 14.6.3, “Importing user data into new Library Server” on page 395

At this stage, you should have completed the following steps:

- Installed a Content Manager Version 8.3 system on AIX, as well as any AIX
  Resource Managers that you wish to migrate to from Windows Object
  Servers.
- Cleared the staging area of your earlier Content Manager Object Server and
  stopped the earlier Content Manager Library Server and Object Server.
- Copied the migrate directory to your earlier Windows Content Manager
  Library Server machine and ran the Migration Wizard against your AIX
  Content Manager Version 8.3 Library Server. This should have generated
  your Library Server folder and Object Server folder(s), and should have
  written the files necessary for the next part of the migration process to these
  folders.
Copied the migrate directory to any location on your AIX Version 8.3 Library Server and any extra AIX Resource Managers. Copied the contents of the folder that the import utility generated from your earlier Content Manager Library Server to your *AIX* Version 8.3 Library Server. Be careful to copy the files over to your AIX machine in such a way that they are unaltered during the copy process. In our scenario, we use an FTP client, set to the *ASCII format*, and copy the files from our Windows Content Manager Version 7.1 Library Server to AIX.

Run the *icmimpl* command from a command line on your AIX Content Manager Version 8.3 Library Server. You must use root, and run the command from the DB2 directory where the Library Server files are copied to. If you have problems with this command, make sure the current directory is in your PATH variable, and that you have read and write permissions on all of the files you copied over from your Windows Library Server. If you experience problems while running any of the migration utilities on *AIX*, most likely, it is a permissions problem.

You should now be able to log into your Content Manager Version 8.3 AIX Library Server using a Windows Client and be able to search through your meta data within the item types. Remember you are not able to retrieve any objects at this time since we have not performed the Objects Server migration yet.

### 14.7.1 Migrating user data from Windows into Version 8.3 RM on AIX

This step is similar to the steps outlined in 14.6.4, “Migrating user data into new Resource Manager(s)” on page 403. Specifically, read and follow the instructions in the following subsections carefully before attempting the migration process:

- “Migrating to a Resource Manager on a different machine” on page 407
- “Copying objects” on page 410

The only difference is that the *icmmidr* executable for Windows platform is the *icmxmidr* command on AIX. This can be seen by editing the icmimpo script on your AIX machine, and looking at the bottom section of the script (see Figure 14-39).
Figure 14-39  Contents of icmimpo script file on AIX

For the AIX scenario, you must first run the icmnmidr command on Windows, then copy the files generated from this command to the DB2 directory on your AIX Version 8.3 Library Server machine (the DB2 directory from the migrate directory that you copied to your AIX machine earlier), and finally run the icmxmidr script on AIX.

Here is the command we run on the AIX server:

```
icmxmidr R rmdb rmadmin rmadmin
```

Once you successfully run the icmxmidr script on AIX to import your Object Server data into your Resource Manager database tables, you need to copy over your LBOSDATA directory. We accomplish this by using an FTP client running in binary mode. Copy the entire LBOSDATA directory from the location on your Windows Object Server to the location on your Version 8.3 AIX Resource Manager. The destination location is the object storage location you specified when installing the AIX Resource Manager.

You LBOSDATA directory may be very large. You need to consider which method of copying the files is the most efficient way for your particular environment. Because we simply copy files over, this does not affect objects
stored within the current Object Server. If something goes wrong with the object copy, we can restart the copying process. While you are copying files from one system to another, you must make sure that no new files are stored into the existing Content Manager system; otherwise, the new Resource Manager database and the objects within its LBOSDATA directory may not be synchronized.

After you copy the LBOSDATA directory and all of its files to AIX Resource Manager, it is important to check the permissions and ownership of the object files and directories. They should be as follows:

► For the LBOSDATA directory itself, and any subdirectories, the permissions and ownership should be:
  
  drwxr-sr-x root:sys

► For all the object files within these directories, the permissions and ownership should be:
  
  -r--r--r-- root:sys

If you currently use TSM on your Object Server, you need to install the TSM Client APIs on your new Resource Manager server and configure the client options file as it is on your original (current) Objects Server. Once this is done, your Version 8.3 Resource Manager can communicate with your TSM server and store and retrieve objects just as your Version 7.1 (or Version 6.1) Object Server did.

There are other methods to move a Resource Manager to a different physical server after the upgrade to Content Manager Version 8.3 is performed. For example, you can create a new Resource Manager on the target machine, then create a storage class on your existing Resource Manager, and configure it so that it specifies your new Resource Manager as a remote destination.

After doing this, you can update your existing migration policy, specifying this new storage class as the next migration step. You may need to update your existing migration definitions if the retention period is set to “forever”, as this means the objects never migrate to the next storage class. Specify all new object stores to go to the new Resource Manager, and when all of the existing objects are migrated to the new Resource Manager, you can delete your original Resource Manager.

**Important:** The final part of the migration process is system validation. You must validate your new system to make sure everything is completed successfully.

Refer to 14.9, “Post migration validation” on page 416 for details.
14.8 Migrating CM Version 7.1 to CM Version 8.3 on AIX

To perform a migration from an existing Content Manager Version 7.1 (or Version 6.1) system running on AIX to a new Content Manager Version 8.3 running on the same machine (you may have Resource Managers on other machines), you should follow the steps in the 14.6, “Migrating from one Windows machine to another” on page 370, because the steps for Windows and AIX are identical.

You also need to decide whether to upgrade the Object Servers to the Resource Managers on your existing machines which is officially supported method, or to move the Object Servers to the Resource Managers on different machines during the migration which is not officially supported. Based on this decision, you need to refer to the relevant sections, “Migrating to a Resource Manager on the same machine” on page 403 or “Migrating to a Resource Manager on a different machine” on page 407.

14.9 Post migration validation

Prior to migration, you should have a set of testing suites which you used to validate that your current system is operating correctly. Make sure that you perform the same set of testing suites on your new Content Manager Version 8.3 system after the migration to validate that everything is migrated successfully and the system is operating as expected. If you have not reviewed all of the log files generated during the migration process, this is a good opportunity to go back and check these log files to ensure that no errors occurred during the migration process.

Once you validate that the new Content Manager Version 8.3 system is operating correctly and that all the existing data is successfully migrated, you can remove the previous version of Content Manager. This includes uninstalling the software code, manually dropping the DB2 databases for the Library Server and the Object Servers, and manually removing any directories left over by the uninstall method that you used. If you want to keep your existing system, you can remove the table space(s) that you created during migration as they are not needed by either systems after the migration.

We strongly recommend that you take a full system backup of your newly migrated Content Manager Version 8.3 system, which includes the Library Server and all the Resource Manager databases. We also recommend that after the migration is complete, you optimize your Content Manager Version 8.3 Library Server and Resource Manager databases (see 18.2, “Optimizing server databases” on page 472 for instructions. Remember to take full database backups before you optimize any database!
TSM migration

In this chapter, we cover the migration of an existing TSM (ADSM) server from Version 3.1.2.1 and higher to Version 5.1.5, as part of a Content Manager migration. We provide steps for both Windows and AIX migrations.
15.1 Migrating from ADSM 3.1.2.1 & above to TSM 5.1.5

Content Manager V7 support Tivoli Storage Manager (TSM) V3.7 and above. If you are upgrading an existing Content Manager V7 system to V8.2, there is a good chance you use either TSM V3.7, V4.1 or V4.2 at the moment and you need to upgrade it to TSM V5.1.5. Content Manager V6 supports ADSM V3.1.2.1 and above. If you are upgrading an existing Content Manager V6 system to V8.2, it is possible you may need to upgrade from ADSM V3.1.2.1 to TSM V5.1.5. For these reasons, we discuss the procedures for upgrading ADSM V3.1.2.1 and above to TSM V5.1.5.

Content Manager V8.2 supports TSM V4.2.1 and above, and is currently shipped with TSM V5.1.5. Even if your current Content Manager V7 system uses TSM V4.2.1 or above, and therefore after upgrading to Content Manager V8, you would still be using a Content Manager supported level of TSM, upgrading TSM to the latest version available is still advisable. By upgrading to a newer version of TSM, it ensures that you are using a supported version of the product, and that you are using a supported version for a longer period of time.

If you are currently using TSM V4.2.1 or above, it is possible to upgrade Content Manager to V8.2 before upgrading TSM; otherwise, TSM (or ADSM) should be upgraded before Content Manager. A TSM upgrade should be a fairly simple procedure; it may be done during the same period you are doing the Content Manager migration.

If you are planning to move an Object Server to a different physical machine when you upgrade to a Content Manager V8.2 Resource Manager, and you currently use TSM, all you need to do is install the TSM Client APIs (Version 5.1.5 or above) on the target server and copy over your existing dsm.opt client options file definitions (and dsm.sys if AIX). Once your TSM server is upgraded, Content Manager can store and retrieve objects from TSM once again.

If your TSM server is currently installed on the same physical machine as your Object Server and you wish to move both servers to different physical machines, you can follow the Object Server to Resource Manager migration guide in “Migrating to a Resource Manager on a different machine” on page 407; for the TSM server move, consult the Tivoli documentation, as this is outside the scope of this redbook. In this scenario, it is possible to move the Object Server to another physical machine, upgrade it to a Resource Manager, and leave the TSM server on the machine that the Object Server used to reside on. In this case, you would simply need to follow the steps in the paragraph above: Install the TSM Client APIs (Version 5.1.5 or above) on the new Resource Manager machine, copy over the client option definitions from the old Object Server machine, and then upgrade the existing TSM server.
The procedures in the following sections cover how to migrate TSM (and ADSM) from V3.1.3.1 and above to TSM V5.1.5, for both Windows and AIX platforms. A Sun Solaris upgrade is not included due to the fact that Content Manager, prior to V8, did not support this platform. These steps should be used in conjunction with the following manuals:

- IBM Tivoli Storage Manager for Windows - Quick Start, GC32-0784
- IBM Tivoli Storage for AIX - Quick Start, GC32-0770

These manuals provide details on how to install TSM, which is a step needed as part of a migrate install.

## 15.2 Migrating to TSM V5.1.5 on Windows

When Tivoli Storage Manager (TSM) Version 5.1.5 is started over a database that was written by ADSM or a previous version of TSM, the database is automatically upgraded.

Before performing an upgrade, it is advisable to perform a full TSM database backup. Refer to Tivoli Storage Manager Administrator's Guide for your current version of TSM for details on how to perform this backup. Also, save a copy of the volume history and device configuration files. For additional information, see the BACKUP VOLHISTORY and BACKUP DEVCONFIG commands in the Administrator's Reference for your current version of TSM.

**Note:** You cannot restore a prior version's backed up database onto the latest version of TSM Server. For instance, you cannot restore a TSM V4.2 database onto TSM V5.1 server.

### Device support

Before migrating your TSM server, ensure that the new server provides support for your current storage devices. Refer to the device support section of the TSM Web site at:


Migrating TSM on Windows can be broken down into the following scenarios:

- “Migrating from TSM V4.2.x.x” on page 420
- “Migrating from TSM V4.1.x.x” on page 420
- “Migrating from TSM V3.7.x.x or ADSM V3.1.x.x” on page 420
**Migrating from TSM V4.2.x.x**

Install the new version of TSM (see *IBM Tivoli Storage Manager for Windows - QuickStart, GC32-0784* for installation instructions) and accept the default path, which points to the existing server location. The setup program automatically upgrades your TSM database and administrative Web interface.

**Migrating from TSM V4.1.x.x**

You need to do the following steps:

1. Write down the directory path of your current TSM server.
2. Use the Windows Add/Remove Programs dialog to uninstall the current TSM server.
3. Install the new TSM server to the same location as the original version (see *IBM Tivoli Storage Manager for Windows - Quick Start, GC32-0784* for installation instructions). The setup program automatically upgrades your TSM database and administrative Web interface.

**Migrating from TSM V3.7.x.x or ADSM V3.1.x.x**

You need to do the following steps:

1. Write down the directory path of your current TSM server.
2. Use the Windows Add/Remove Programs dialog to uninstall the current TSM server.
3. Install the new TSM server to the same location as the original version (see *IBM Tivoli Storage Manager for Windows - Quick Start, GC32-0784* for installation instructions).

   After you complete the installation, the Initial Configuration Task List dialog appears. Close this dialog.

4. Update and run the script tsmfixup.cmd, located in the console directory. (Refer to the script header for update instructions.) The script automatically upgrades your TSM database and administrative Web interface.

   If you are migrating from ADSM and have created a disaster recovery plan file using Disaster Recovery Manager (DRM), be aware that TSM does not use the same default installation directories as ADSM. Disaster recovery installation path references may no longer be valid.
After you migrate to TSM, you should back up your storage pools and database, and create a new disaster recovery plan file (if this is a feature you used in the old version of the product). For the sequence and details of the procedure, refer to the Tivoli Disaster Recovery Manager chapter in *IBM Tivoli Storage Manager for Windows - Administrator's Guide*, GC32-0782.

15.3 Migrating to TSM V5.1.5 on AIX

It is possible to install Tivoli Storage Manager (TSM) Version 5.1.5 over a previous version of ADSM or TSM. This is called a “migrate install”. A DSMSERV UPGRADEDDB operation is automatically performed during a migrate install.

The ADSM default installation directories changed for TSM. If you have previously used Disaster Recovery Manager (DRM) to create a disaster recovery plan file, that file refers to path names that may no longer be valid. After you install the TSM, you should back up your storage pools and database and create a new disaster recovery plan file (if this is a feature you used in the old version of the product). For the sequence and details of the procedure, refer to the Tivoli Disaster Recovery Manager chapter in *IBM Tivoli Storage Manager for AIX - Administrator's Guide*, GC32-0768.

To return to ADSM or an earlier version of TSM, after a migrate install, you must have a full database backup of that original version and the install code for the server of that original version.

**Note:** You cannot restore a prior version's backed up database onto the latest version of TSM Server. For instance, you cannot restore a TSM V4.2 database onto TSM V5.1 server.

Be aware of the consequences of returning to an earlier version of the server:

- References to client files that were backed up, archived, or migrated to the TSM V5.1 server will be lost.
Some existing volumes may be overwritten or deleted during the TSM V5.1.5 operation. If so, client files that were on those volumes and that were migrated, reclaimed, moved (MOVE DATA command), or deleted (DELETE VOLUME command) may no longer be accessible to the earlier version of the server.

Definitions, updates, and deletions of objects performed on the Version 5.1.5 server will be lost.

**Migrating to AIX Version 5.1**

Device driver conflicts occur if you have TSM V4.2 installed on AIX Version 4.3.3 and want to migrate to AIX Version 5.1. To resolve this, the TSM device support for AIX Version 5.1.5 (tivoli.tsm.devices.aix5.rte) must be installed. To do this, follow these steps:

1. Before migrating, record all TSM device definitions.
2. Uninstall the fileset tivoli.tsm.devices.aix43.rte by using the command:
   ```
   installp -ug tivoli.tsm.devices.aix43.rte
   ```
   This will also cause tivoli.tsm.msg.[lang].devices to be uninstalled, where [lang] is en_US and any other “tivoli.devices” messages fileset that may be installed.
3. Migrate to AIX Version 5.1
4. Install the filesets tivoli.tsm.devices.5.rte and tivoli.tsm.msg[lang].devices.
5. Redefine the devices.

15.3.1 **Before performing a migrate install**

This section describes some of the things you should consider before you perform a migrate install:

- The dsmserv.dsk file points to the locations of the current database and recovery log volumes. A migrate install does not normally create a new database, recovery log, and storage pool volumes. If, however, dsmserv.dsk is not in the `/usr/lpp/admserv/bin` or `/usr/tivoli/tsm/server/bin` directory, the install creates the following volumes in the `/usr/tivoli/tsm/server/bin` directory:
  - Database volume (db.dsm)
  - Recovery log volume (log.dsm)
  - Storage pool volumes (backup.dsm, archive.dsm, and spcmgmt.dsm)

To use your existing database, recovery log, and storage pool volumes, ensure that a copy of the dsmserv.dsk file is in `/usr/tivoli/tsm/server/bin` and the file system is mounted before you do a migrate install. You must not move the database, recovery log, and storage pool volumes.
If you decide to return to the previous version of the server, you must have a backup copy of your prior database, volume history, and device configuration files. In the following example, the tape device class named TAPECLASS is used for database backups:

```
backup db type=full devclass=tapeclass
backup devconfig filenames=devconfig.Sep03
backup volhistory filenames=volhistory.Sep03
```

**Note:** This command retrieves database records into a database dump. This process does not access the recovery log. Transactions held in the log database are lost.

Store the output volumes and the device configuration and volume history files in a safe location.

During the migrate install, the following files are automatically copied to the location of the new TSM installation:

- The dsmserv.dsk file
- The accounting log file (dsmacct.log)
  
  If the environment variable DSMSERV_ACCOUNTING_DIR was set, you must reset it.
- The server options file dsmserv.opt
- Existing device configuration and volume history files if they are named devconfig and volhist

**Important:** If these files are not named devconfig and volhist, you should back up the files and save them in a temporary directory. You can later edit the new server options file to include the names of these files.

If the files are automatically copied, the server options file is automatically updated.

You should also save in a temporary directory any existing runfile scripts.

- TSM device definitions are not saved during a migrate install. To install the new TSM drivers, you must have the output from the following commands:

```
lsdev -Cc tape
lsdev -Cc library
```

This is not required for the IBM 3494, 3495, 3570, 3575, or 3590, which use drivers supplied with the devices.
15.3.2 Performing a migrate install on AIX

To perform a migrate install, do the following steps:

1. Stop the server if it is running by entering:

   ```
halt
   ```

   If you started the server as a background process, connect to the server as an administrative client and issue the HALT command. If you cannot connect to the server with an administrative client, you must use the `kill` command with the process ID number (PID) that is displayed at initialization.

   **Note:** Before you issue the `kill` command, ensure that you know the correct process ID for the server.

2. If you have created scripts that have paths to `/usr/lpp/admserv/bin`, change the paths to `/usr/tivoli/tsm/server/bin`.

3. Install the latest TSM server software (see *IBM Tivoli Storage Manager for AIX - Quick Start, GC32-0770* for installation instructions).

   **Note:** Until the new version is installed and any required licenses are reregistered, clients are not able to connect to the server.

4. TSM is shipped with sample command scripts that can be loaded into the database and run from an administrative client, administrative Web interface, or server console. They can also be included in administrative command schedules. The sample scripts, in scripts.smp, are primarily SELECT queries; but also include scripts that define volumes for and extend the database and recovery log and that back up storage pools.

   **Note:** The sample scripts may have been loaded when a previous version of ADSM or TSM was installed. Loading the sample scripts again at this point will overlay any existing scripts of the same name; any modifications made previously to those scripts will be lost.

   To load the sample scripts into the database, issue the following command:

   ```
   ./dsmserv runfile /usr/tivoli/tsm/server/webimages/scripts.smp
   ```
5. To use the Web administrative interface:
   a. Your browser must provide Java 1.1.6 support.
   b. Configure the HTTP communication method in your server options file
      (dsmserver.opt).
      
      commmethod http
      httpport 1580

6. Start the server.

   . /dsmserver

7. Your licenses from the previous version are no longer valid and must be
   reregistered.

**Note:** The tivoli.tsm.license package is required to register licenses. This
package is installed when you install the server package.

### 15.4 Post migration steps

After completing a TSM server migration on either Windows or AIX, it is very
important to verify that the migration is successful. A simple way to accomplish
this, using Content Manager Windows client, is to retrieve an object that you
know is stored on media controlled by TSM, but not anywhere else.

This tests the communication between Content Manager and TSM, verifying that
your TSM client options files are configured correctly, your ICMRM.properties file
is configured correctly, and that the TSM policy structure that was set up in the
earlier version of TSM is still working as expected. This test also verifies the
TSM’s device definitions, which enable the TSM server to communicate with
external storage devices through its device drivers.
In this chapter, we discuss special migration scenarios which are not officially supported by the standard IBM Content Manager migration utilities. They include cross-platform migration from z/OS to AIX or vice versa, migration from a third-party vendor product to Content Manager, and merging of multiple Library Servers into one.

Our intention is to provide the approach for various special migration scenarios. In certain scenarios, we provide detailed step-by-step instructions on how to achieve the migration. They should be used “as-is” and none of these are officially supported by IBM.
16.1 Migration paths

Due to your business requirements, there are different ways you can migrate an existing system to the new Content Manager Version 8.3 system. Table 16-1 lists the main possible migration paths.

Table 16-1 Migration paths

<table>
<thead>
<tr>
<th>From \ To</th>
<th>Windows</th>
<th>AIX</th>
<th>z/OS</th>
<th>Other product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIX</td>
<td>N/A</td>
<td></td>
<td>Chapter 16.3, on page 442</td>
<td>N/A</td>
</tr>
<tr>
<td>z/OS</td>
<td>N/A</td>
<td>Chapter 16.2, on page 430</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Other product</td>
<td>Chapter 16.6, on page 445</td>
<td>Chapter 16.6, on page 445</td>
<td>Chapter 16.6, on page 445</td>
<td>N/A</td>
</tr>
</tbody>
</table>

We cover the version-to-version (Version 6.1 or Version 7.1 to Version 8.3) Content Manager migration from Windows platform to Windows and from AIX to AIX.

We also cover the Content Manager migration from a Windows-based system to an AIX-based system.

In the following sections, we cover cross-platform migration from z/OS to AIX and vice versa. We also discuss migration from other vendor products to Content Manager.
In addition to the main migration paths listed in Table 16-1, there is a special migration scenario that we also address in this chapter: how to merge two Library Server into one. This is a situation which can arise when two companies are merging and the existing two systems need to be consolidated. We cover the approach for this scenario in a very high level to give you an idea how to go about merging them.

Figure 16-1 shows the merging of two Library Servers.
16.2 Migrating from CM for z/OS V8.3 to CM for MP V8.3

In this section, we cover the Content Manager (CM) migration of the Library Server from z/OS to Multiplatforms (MP) platform. Although we only cover the migration from z/OS to an AIX platform, this process is applicable for a migration from AIX to z/OS, and for a migration between z/OS and Windows platforms.

With Content Manager Version 8.3, the Library Server has an identical interface and layout for both Multiplatforms and for z/OS. This is true for the DB2 table structures, stored procedures and functions. In fact, this has been the case since Version 7.1; but since there was never an announcement for a Content Manager Version 7.1 for z/OS, this was not a well known fact. With the announcement of Content Manager for z/OS Version 8.2 and now for Version 8.3, it is now well documented.

**Note:** There is one extra table in the z/OS environment which can be used to define additional clauses to the SQL create statements. For the migration purpose, this is not relevant.

With Content Manager Version 8.3, the Resource Manager on Multiplatforms and on z/OS have totally different designs. Not only they are different in database structure, they are also different in the way the objects are stored and managed. Also, there is no remote migration available from z/OS to AIX.

A Resource Manager migration is a completely different task that is beyond the scope of this redbook and thus it is not covered.

16.2.1 Migration process overview

Figure 16-2 shows the general migration flow when migrating from a Library Server on z/OS to AIX or vice versa.
Figure 16-2  General migration flow

As shown in Figure 16-2, the migration process includes the following steps:

- Step 1: Install the Library Server on the target system.
- Step 2: Drop referential integrity constructs (RIs) of Library Server database.
- Step 3: Unload the data from the source system.
- Step 4: Move the data to the target system.
- Step 5: Import data onto the target system.
- Step 6: Create referential integrity constructs (RIs) onto the target system.
- Step 7: Perform bind.
- Step 8: Test target system.

We address each step in detail in the sections starting from 16.2.3, “Step 1: Install Library Server on target system” on page 432.

**Important:** The migration process we describe here is from the result of practical lab exercises performed when writing this redbook. The described method of moving a Library Server from one system platform to another is not officially supported by IBM.

You should use the information on an “as-is” basis. We provide this information to give you a general understanding of what to do when you encounter this migration requirement. You should contact an IBM service representative if you need to perform this type of migration.
16.2.2 Migration consideration and preparation

Prior to addressing each step in the migration process flow, you need to consider all the migration related issues, and to plan and prepare for the migration. The issues you need to consider include available disk space, system backup, and available time for migration process. Although we are only migrating Library Server database, the files generated from unloading of the Library Server data can easily go up to gigabytes. This, in turn, may have impact on the time needed to run the migration task. Some of the questions you should ask yourself include:

- Do you have enough disk space to do this migration?
- Do you have enough time to do it without impacting the production system?
- If you are doing this during the prime time, how will this impact the production system?
- Do you have a set of test suites to validate your migration?
- How long does it take you to validate that the migration is successful?

Make sure you ask enough questions, plan ahead for the possible problems, and know how to handle them.

16.2.3 Step 1: Install Library Server on target system

This involves installing Library Server to your target system and setting up all the necessary environment variables. This includes setting up the correct path to your stored procedure programs which is stored in the ICMSTSysControl table.

**Important:**
- The user ID to install the Library Server must be the root user ID.
- The database schema of the Library Server database on AIX must be the same as on z/OS.
- The database name of the Library Server database on AIX must be the same as on z/OS.
- The user ID used as administration user on the z/OS must be a DB2ADMIN user on the AIX machine.

After the database is created, run the `db2look` command to create the DDL for this database from the DB2 command window:

```
db2look -d <db name> -e > ls.ddl
```

Where:
- `<db name>` is the Library Server database.
- `> ls.ddl` pipes the output to `ls.ddl`. You can use any name for the output file.
The output file (in our scenario, ls.ddl), is a DDL file for the database, including all the table spaces, tables, index, and referential integrity information for that database. The output is stored in a plain text file and can be used as input for other commands. In our scenario, we only use the last part of the output file, starting after the word “foreign” (see Example 16-1). When you perform a search for the word “foreign” in the output file, you can find all the referential integrity construct definitions. These definitions must be rebuild in a later step of the migration. For this purpose we used this part of the output file of **DB2LOOK**.

**Example 16-1  DDL file sample**

```sql
-- DDL Statements for foreign keys on Table "ICMADMIN"."ICMSTPRIVSETS"

ALTERTABLE "ICMADMIN"."ICMSTPRIVSETS"
  ADD CONSTRAINT "SQL020802165247620" FOREIGN KEY
  ("PRIVSETCODE")
  REFERENCES "ICMADMIN"."ICMSTPRIVSETCODES"
  ("PRIVSETCODE")
  ON DELETE CASCADE
  ON UPDATE NO ACTION;
```

### 16.2.4 Step 2: Drop referential integrity constructs of LS database

This involves dropping of the referential integrity constructs (RIs) of the Library Server (LS) database. Without the referential integrity constructs, you can load the data with the **DB2MOVE IMPORT** utility; otherwise, you get errors during the import.

To generate a list of the existing RIs, perform the following DB2 select statement against the database from the DB2 command window:

```
db2 select tbname, relname, refkeyname from sysibm.sysrels > sysrel.txt
```

You should get a list of the existing RIs for the Library Server database. Example 16-2 provides a sample output of the select statement, with table name (TBNAME), referential integrity (RELNAME), and reference column name (REFKEYNAME).

**Example 16-2  sysrel.txt - Sample output of select statement for existing RIs**

<table>
<thead>
<tr>
<th>TBNAME</th>
<th>RELNAME</th>
<th>REFKEYNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMSTPRIVSETS</td>
<td>SQL030925164851180</td>
<td>ICMSTPRIVSETCODES</td>
</tr>
<tr>
<td>ICMSTPRIVSETS</td>
<td>SQL030925164851210</td>
<td>ICMSTPRIVDEFS</td>
</tr>
<tr>
<td>ICMSTPRIVGROUPS</td>
<td>SQL030925164852210</td>
<td>ICMSTPRIVGROUPCODECODE</td>
</tr>
<tr>
<td>ICMSTPRIVGROUPS</td>
<td>SQL030925164852230</td>
<td>ICMSTPRIVDEFS</td>
</tr>
<tr>
<td>ICMSTDOMAINPRIVSET</td>
<td>SQL030925164852790</td>
<td>ICMSTADMINDOMAINS</td>
</tr>
<tr>
<td>ICMSTDOMAINPRIVSET</td>
<td>SQL030925164852810</td>
<td>ICMSTPRIVSETCODES</td>
</tr>
</tbody>
</table>
Save the list in a file. In our scenario, we pipe the select statement output to the file, sysrel.txt. Edit the file so that it includes the drop statements for all the RIs. In our scenario, we edit the file and save it to RIDrop file. See Example 16-3 for our sample RI drop file.

**Example 16-3  RIDrop - Sample RI drop file**

| Statement                                                                                                    |
|---|----------------------------------------------------------------------------------------------------------------|
| ALTER TABLE ICMSTPRIVSETS DROP CONSTRAINT SQL030925164851180;                                              |
| ALTER TABLE ICMSTPRIVSETS DROP CONSTRAINT SQL030925164851210;                                              |
| ALTER TABLE ICMSTPRIVGROUPS DROP CONSTRAINT SQL030925164852210;                                             |
| ALTER TABLE ICMSTPRIVGROUPS DROP CONSTRAINT SQL030925164852230;                                             |
| ALTER TABLE ICMSTDOMAINPRIVSET DROP CONSTRAINT SQL030925164852790;                                         |
| ALTER TABLE ICMSTDOMAINPRIVSET DROP CONSTRAINT SQL030925164852810;                                         |

Run the file (RIDrop) to drop all the referential integrity of the database.

### 16.2.5  Step 3: Unload data on source system

We use the **DB2MOVE EXPORT** command to unload the data from the z/OS DB2. This utility gets the data and the database definition from the source z/OS database and stores it, together with the restoring information, in files.

We run the utility from a Windows workstation. You must connect this workstation with the z/OS DB2. The **DB2MOVE** command is also available on AIX. Since we have a small database and we do not need to consider system performance, we decide to use a Windows workstation to perform this task.

Execute the **DB2MOVE EXPORT** command from DB2 command window as follows:

```
db2move <db name> export -tc <table-creators> -tn * -sn <schema-names> -u <user ID> -p <password>
```

Where:

- `<db name>` is the Library Server database.
- `<table-creators>` is the creator of the tables.
- `<schema-names>` is the schema name of the Library Server database.
- `<user ID>` is user ID with administration rights on the Library Server Database.
- `<password>` is the valid password of the user ID.

Example 16-4 shows what happen when we execute the **DB2MOVE EXPORT** command.
Example 16-4  DB2MOVE utility output

*****  DB2MOVE  *****

Action: EXPORT

Start time: Fri Sep 26 14:54:47 2003

Exporting tables created by: ICMADMIN;

All table names beginning with: *

Server: DB2 for MVS V7.1.1

EXPORT: 4 rows from table "ICMADMIN"."ICMUT0100001"
EXPORT: 4 rows from table "ICMADMIN"."ICMSTACCESSCODES"
EXPORT: 5 rows from table "ICMADMIN"."ICMSTCOLLNAME"
EXPORT: 1 rows from table "ICMADMIN"."ICMSTACCESSLISTS"
EXPORT: 400 rows from table "ICMADMIN"."ICMSTCOMPILEDACL"
EXPORT: 97 rows from table "ICMADMIN"."ICMSTCOMPILEDPERM"
EXPORT: 121 rows from table "ICMADMIN"."ICMSTATTRDEFS"
EXPORT: 84 rows from table "ICMADMIN"."ICMSTATTRGROUP"
EXPORT: 23 rows from table "ICMADMIN"."ICMSTCOMPDEFs"
EXPORT: 380 rows from table "ICMADMIN"."ICMSTCOMPATTS"
EXPORT: 3 rows from table "ICMADMIN"."ICMSTCOMPATTSFK"
EXPORT: 23 rows from table "ICMADMIN"."ICMSTCOMPVIEWDEFS"

The exported files resulting from the DB2MOVE EXPORT command are in IXF format. The IXF format is a preferred DB2 interchange format. When using this, we do not have to perform additional data conversion (such as from EBCDIC to ASCII), because this is done automatically.

Tip: To estimate the size of the output-IXF files, you should run a test with your real data. We do not provide any formulas here, since there are too many dependencies that can affect the calculation and we do not have a way to accommodate them.

Note: The user ID which is connected to the host database needs to have the correct level of authority to bind plans (BINDADD, BIND)

16.2.6 Step 4: Move data to target system

We do not have to move the data, because we are running the DB2MOVE IMPORT from the same workstation. Within this Windows workstation, we connect it to the z/OS system, unload the data, connect it to the AIX system, and run the import routine directly from the workstation.
If you have a large database, for performance reasons, you may want to bring the data to the target system, and then run the import routine.

### 16.2.7 Step 5: Import data onto target system

To import data onto the target system, use `DB2MOVE IMPORT REPLACE_CREATE` command. With this option, during the data import, the tables that are not already defined on the Library Server database are created automatically.

Execute the `DB2MOVE IMPORT` command from DB2 command window as follows:

```bash
db2move <db name> import -io replace_create -u <user ID> -p <password>
```

Where:
- `<db name>` is the Library Server database.
- `<user ID>` is user ID with administration rights on the Library Server Database.
- `<password>` is the valid password of the user ID.

**Important:** Before starting with the data import, remove the control statement for table ICMSTSysControl from the `DB2MOVE` control file.

After you create and load the tables, it may be necessary to update the ICMSTSysControl table of your Library Server database. Compare the values from your z/OS table with the one on the AIX server. Do not touch the path entries and the decoded columns.

You may find a difference in the System Flag column, which is set to 32 on the AIX. This is correct.

Make sure that the Library Server ID on your AIX is set to the same value as on the z/OS. Otherwise nothing will work.

### 16.2.8 Step 6: Create referential integrity constructs of LS database

After successfully loading the data into the tables, you need to rebuild the referential integrity constructs (RIs) of the Library Server (LS) database. To do this, use the DDL generated by the `DB2LOOK` after installing the Library Server database.

**Note:** Run only the last part of the DDL which generates the RIs. Search for “foreign” in the DDL to find where this section begins.
Example 16-5 shows a sample file, RICrt, used to create referential integrity.

**Example 16-5   RICrt - Sample RI create file**

-- This CLP file was created using DB2LOOK Version 7.2  
-- Timestamp: Mon Sep 29  20:34:16  2003  
-- Database Name: MVSLSDB  
-- Database Manager Version: DB2/6000 Version 7.2.8  
-- Database Codepage: 819  

-- DDL Statements for foreign keys on Table "ICMADMIN"."ICMSTPRIVSETS"

ALTER TABLE "ICMADMIN"."ICMSTPRIVSETS"  
  ADD CONSTRAINT "SQL030929202519390" FOREIGN KEY  
       ("PRIVSETCODE")  
       REFERENCES "ICMADMIN"."ICMSTPRIVSETCODES"  
       ("PRIVSETCODE")  
   ON DELETE CASCADE  
   ON UPDATE NO ACTION;

ALTER TABLE "ICMADMIN"."ICMSTPRIVSETS"  
  ADD CONSTRAINT "SQL030929202519430" FOREIGN KEY  
       ("PRIVDEFCODE")  
       REFERENCES "ICMADMIN"."ICMSTPRIVDEFS"  
       ("PRIVDEFCODE")  
   ON DELETE RESTRICT  
   ON UPDATE NO ACTION;

To run this RICrt file, use the following DB2 command in the DB2 command window:

```
db2 -xvf RICrt
```

**16.2.9 Step 7: Bind plans**

First, you have to bind the plans for the DB2 Connect. Use the DB2 Client Configuration Assistant to bind the plans.

You then need to bind the Content Manager plans on AIX. Use the script that comes with the installation. The bind script name is icmbindlsdb.sh. It is located in the /usr/lpp/icm/config directory.
Figure 16-3 shows the start of the binding of the plans for the DB2 Connect.

Figure 16-3  Bind start

Figure 16-4 shows the next step in binding the plan for the DB2 Connect.

Figure 16-4  Bind next
Figure 16-5 shows the bind results of the plan for the DB2 Connect.
Figure 16-6 shows the screen capture when binding the Content Manager plans on AIX.

![AIX terminal output](image)

Figure 16-6  Bind the Content Manager plans

### 16.2.10 Step 8: Test target system

You need to validate the migration process by testing the AIX Library Server that is connected to the z/OS Resource Manager. Before you can view an object, you have to send a new encryption key to the Resource Manager. After this action, you can store and retrieve new objects, and also retrieve the objects which have been stored on the z/OS system earlier.

To set a new encryption key, do the following steps:

1. Start the System Administration Client.
2. Expand the Library Server database (icmnlsdb) → Library Server Parameters → Configurations.
3. From the right-hand panel, double-click Library Server Configuration.
4. Click Refresh encryption key. See Figure 16-7.
You will receive a warning message, saying that it is recommended that you only do this at the least traffic time. The warning message prompts you to continue. Click Yes and the key is refreshed.

There will not be any confirmation message specifying that it is completed; but the encryption key is refreshed.

16.2.11 Summary

There are eight steps involved in the migration of a Library Server from z/OS to AIX (see Figure 16-2 on page 431). When writing this redbook, we tested it with the actual lab exercises and found that this process works. We recommend that you use these eight steps as a general migration process flow. In principal, regardless of what the source system platform is and what the target system platform is, the steps involved are generally the same. They may differ slightly in the setup of the user IDs and user rights, the way the servers are installed, and how to perform tasks such as binding of DB2 plans. You need to work with your system administrators to figure out the details.

Because the process requires mostly standard DB2 tools, it is important to be aware of the current versions of DB2 on your systems. As a general rule, DB2 servers are compatible with the lower versions of the clients; but not vice versa.
For example, Version 8 DB2 server can work with Version 7 DB2 client; however, you cannot use a Version 8 DB2 client with a Version 7 DB2 server.

Note that the database structure of the Content Manager for Multiplatforms and the one for z/OS are really identical. The layout and the data itself are also identical. Otherwise, in our scenario, the migration would not have worked.

When you plan for a migration involving a large Library Server database, pay special attention to the time it may take for you to complete the task and the available disk space. You want to make sure you have adequate time to convert all of them. There may also be some problems that are environment dependent. For example, a decimal point may mean a comma in one system and an actual decimal point in another system. Language conversions may vary as well.

16.3 Migrating from CM for MP V8.3 to CM for z/OS V8.3

The process flow is same as the one described in 16.2, “Migrating from CM for z/OS V8.3 to CM for MP V8.3” on page 430. Read that section and follow the steps covered in the section for your migration.

To get a quick overview of the process flow, refer to Figure 16-2 on page 431 for the eight steps involved.

Important: If you want more information about how to migrate from Content Manager for Multiplatform to Content Manager for z/OS Version 8.3, see the IBM Redbook *DB2 Content Manager for z/OS Version 8.3 Implementation, Installation, and Migration*, SG24-6476, chapters 7, 8, and 9.

16.4 Cross-platform migration: Older version to CM V8.3

In this section, we discuss how you can migrate from an earlier Content Manager system to the new Version 8.3 system that uses a different platform. For example, you may need to migrate a Content Manager for z/OS Version 7.1 system to a Content Manager for Multiplatform Version 8.3 system.

We recommend the following approach:

1. Migrate the data from the source platform to the target platform.

2. Using the official migration tool, migrate the earlier version of Content Manager system (DB2 data only) that is now on the target system to the latest version of Content Manager.
For example, if you need to migrate Content Manager Version 7.1 for z/OS to Content Manager Version 8.3 for AIX, we recommend the following migration path:

1. Migrate Content Manager Version 7.1 data from z/OS to AIX.
2. Upgrade Content Manager from Version 7.1 to Version 8.3 on AIX.

The reasons for this approach are as follows:

- When perform data migration from source platform to a target platform for earlier Content Manager data, it involves fewer database tables. It is therefore easier to perform the data migration. Of course, this is only applicable when dealing with Version 7.1 to Version 8.3 Content Manager cross-platform migration.
- This approach does not require a running version of the latest Content Manager on the source platform; you only need to move the DB2 data to the target system.

**Step 1: Migrate data to the target system**

Follow the process flow described in 16.2, “Migrating from CM for z/OS V8.3 to CM for MP V8.3” on page 430, to migrate data from the source platform to the target platform.

When moving the DB2 data between the systems, you must make sure that you use the correct code page translation, the translation between EBCDIC and ASCII, and the decimal point definition.

For the code page and ASCII conversion, there are tools available. If your definition of the decimal points differs, you may need to write a special utility to convert data.

**Tip:** Use DB2LOOK and DB2MOVE utilities as described in 16.2, “Migrating from CM for z/OS V8.3 to CM for MP V8.3” on page 430 for DB2 data migration. Because you are doing a migration on the database level only, these utilities help you in the data conversion.

**Step 2: Migrate to Content Manager Version 8.3**

When all the data is transferred to the target system and properly stored in the DB2 tables, you can use the standard migration utilities provided by IBM to migrate from an earlier version of Content Manager (data only) to Content Manager Version 8.3. See 14.2, “Upgrade considerations” on page 327 to get information about how upgrade to DB2 Content Manager Version 8.3.
16.5 Merging Library Servers

There is no official support for merging two or more Library Servers to one; however, you may encounter this requirement in the following circumstances:

- **First scenario:** You are preparing for the installation of a new system and there may be a need to merge an existing system to it later.

- **Second scenario:** There are already two or more established systems running and you need to merge them.

**First scenario:** In this case, you especially need to plan for the installation of the new system. For example, you should plan to use different Library Server IDs for your new Library Server. With this, the new system generates different item IDs on the its server for the items. They will not conflict with the existing system's IDs when you merge the existing system into this new one later. Make sure you use different user IDs and different names when defining data model, such as attribute names and item type names.

In addition, you need to find a way to separate the numbers for the newly created item types. If you know that you have to merge an existing system into this new one later, you may create dummy item types on the new system (the system where the other one is to be merged to). These dummy item types can be the place holders for the existing system's item types.

If you already know exactly what item types are to be created from the existing system, you can create them on the new system first. You can load the data from the existing system later. If possible, we recommend defining all the other objects such as collections and Resource Managers from the existing system to the new system.

**Second scenario:** You may not always have this information, or have such a requirement, when a Content Manager system is installed. Sometimes, when two companies, both using Content Manager, are merged into one, there is a need to consolidate the Library Servers. These systems may exist for years. The system definitions and user definitions may or may not overlap.

In this scenario, you may need to go through all the tables manually, checking for the dependencies, and updating your data, so that you do not have any duplicate internal numbers or names.

Use the Information Center of the Content Manager installation to gather all the Content Manager database related information. All the tables and cross references are listed there.
Depending on the size of the existing systems, the second scenario calls for a service project. The relevant information for the project is the number of metadata items defined, such as item types and links, rather than the number of objects in the systems.

If you have a large system, you may consider using the Information Integrator for Content and federated searches on these systems. This may increase the response time; but, it can be implemented in an easy way. Also, there are many experienced developers who can assist with these type of projects. This may enhance the flexibility of the systems.

If the business need mandates that you have only one Content Manager system running, the easiest way to reach this goal is to write an application that reads the documents from one system and stores them into the other. With this approach, all the data have to be moved as well: you need to move all the data from one Resource Manager to another. With this migration path, you merge all the Library Servers to one and you can validate that everything works.

16.6 Migration from third-party products to CM

When migrating from a third-party product to Content Manager, you must decide on the target platform that you want to migrate to, then design the system topology and data model. Once the design is done, we recommend using the seven steps shown in Figure 16-8 as a general guideline to help you in going through your migration.
The migration process flow consists of seven steps:

1. Extract the meta data from the existing system using an export utility from the third-party product.
2. Write the meta data information to a separate database.
3. Read the meta data and the corresponding objects by the import preparation program.
4. Write the reformatted objects and the corresponding meta data to hard disk.
5. Read the XML file with meta data and the file name of the corresponding object.
6. Get the objects.
7. Call the appropriate Content Manager Version 8.2 API to store the object in the Content Manager.

Figure 16-8  Migration flow for migration another product to Content Manager
16.6.1 Planning and preparing for migration

Migrating from a third-party system to Content Manager takes a lot of planning. In addition to the normal project planning, which is beyond the scope of this redbook, we want to emphasize that you must allow ample time to plan ahead. Depending on how many objects need to be transferred and how complex your data model is, the time frame can easily span over a year.

Make sure you think through all the options and issues prior to executing the actual process. Some of the questions you may want to address include:

- If it takes approximately one year to complete the migration, what do you do for the intermediate state?
- How do you keep both the Library Server and the Resource Manager in synchronization?
- How do you continue using your existing production system and the new system at the same time?

Again, keep in mind that you are planning for a long running project.

You must also do the following steps to prepare for the migration:

- Analyzing the source system
- Analyzing the target system
- Describing the migration scenario
- Developing the import program
- Setting up the data transfer procedure
- Setting up the migration environment

**Analyzing the source system**

This includes analyzing the opticals or tapes used in your existing system. It also includes analyzing the data files. You need to see in what format the data is written. There are always differences between the information you get from the documentation and what is really on disks. Often, the data format has changed during the years; therefore, you have to verify what you are dealing with. You must also find a way to extract the meta data of the objects. Usually, the third-party product has a system utility which allows for extraction of the meta data. Try to use the existing utility if possible.

**Analyzing the target system**

It is very important to analyze the target system as well as the source system. You must take in consideration the impact that the data import has to the target system. If you are migrating into an empty Content Manager system, you may only need to ensure enough storage space is available and that the system has enough power to perform the imports in the expected time frame.
If after a while, your target system is expected to go into production while you import additional data from the source system, you need to analyze whether the target system can accommodate this without affecting the system performance. In addition, pay close attention to the network performance. Whenever possible, place the importing workstation and the Content Manager Resource Manager in the same segment.

**Describing the migration scenario**

For step one and two mentioned in Figure 16-2, you need to describe your migration scenario and documenting the process. This is one of the most important things in the project. Thought out all the situations; document the protocols you are going to use to handle each situation during the process.

First of all, decide what you will use as the primary data source. This can be the meta data in the database or the physical data storage, such as the optical disks. Some other application may hold the information about the existing objects; you may use that instead. What you will use depends on the circumstances. We list some questions that may help you to make your decision. For ease of the questions, we assume you use optical disks. Some of the questions you should ask yourself are:

- Is there a backup of all disks available?
- What is the quality of the backup volumes?
- Can the backup volumes be transferred to another place or must they stay on the system site?
- If transferable, how long can they stay off site?
- In case of a crash of the production system, in what time frame must the backup be at the production site again?

Depending on your answers to these questions and other questions you may come up with, you have to decide what to use in your situation.

For some of the customers we work with, they usually have a good backup strategy; their backup volumes are usually in good condition. We, therefore, choose the backup volumes as the primary data source. They can be sent to a sub-contractor who is specialized in reading optical volumes physically, without any knowledge of the meta data. The sub-contractor reads the object data from the volumes, re-formats the data to the data type we want, and connects the object data with the meta data. This can be done using XML files. In this way, you can get the most out of the source system.

**Note:** You will need to figure out how to link meta data to these objects (documents).
Several problems may arise from these two steps. You might have meta data that has no matching objects (documents), or objects without any meta data information. In addition, the objects and the meta data may have conflicting information. For example, a document may have less or more pages than what is specified in the meta data.

These problems need to be reported, documented, and handled. A migration cannot simply be done by counting the number of objects (documents) in the existing system and the number of migrated objects in the new system. These two numbers may never match. After a period of operation, a production system may no longer be in synchronization with itself. Watch out for the potential problems and figure out how you can handle them.

**Developing the import program**

The next step (the third one, as referenced in Figure 16-2 on page 431) is to develop an import program. The program must perform the import from the intermediate stage of the data (from step two) into the Content Manager system. You can use a Windows program to do this.

For performance reasons, your program should be able to run several instances at the same time. When testing your process, find out how many instances you can run simultaneously for how long. This is mainly influenced by the available system resources. You can use this information to better estimate the actual length of the import process.

**Setting up the data transfer procedure**

If you can send the data to be migrated to the site where you have your migration tools running, you have to set up a transfer procedure. Again, you need protocols. If you cannot send the data out of the production site, you need to build the migration environment at the production location. Keep in mind that you may need the space and resources for a long time. Plan ahead.

**Setting up the migration environment**

Next is setting up the migration environment. This includes the installation of the import machine. When you specify the machine for the import, look at how the imported data is sent to you. The import machine must read these files. Try to avoid network communication when reading these files. Whenever possible, put the hard drive in the machine itself.
16.6.2 Performing migration

When you have completed the previously mentioned steps and other necessary procedures, you can start the migration on the production system. During the first few days, you should carefully monitor the system resources, performance, and response times. Depending on the system performance, you may need to tune the system to get better performance.

**During the migration process**

As we have already mentioned, the migration may take a long time. It is therefore important to set some milestones where you can check the status of the project. During the normal import activity, usually, there are few problems; however, you need to routinely check to ensure that everything is going as expected. We recommend performing regular cleanup of the system and environment. Depending on the time frame and the volume of your migration, as an example, this can be every one or two months.

16.6.3 Post migration cleanup

When the migration is finished, it is important to clean up all the intermediate files. Delete anything that is no longer required from the migration machines. Make sure to get back all the disks and volumes if you asked a sub-contractor to do the data conversion.
Application migration

In this chapter, we offer advice on how to port your existing IBM Content manager Version 6 and Version 7 Applications to IBM Content Manager Version 8.3.

The intent is to provide an entry point into application development on IBM Content Manager Version 8.3. It is not intended to provide a full and complete breakdown of programming in an IBM Content Manager environment. To understand the concepts involved in application development, refer to Chapter 6, “Application development overview” on page 131.
17.1 Introduction

Migrating your applications can be a lengthy and complicated process. It requires preparation and planning before undertaking this task. You should migrate a custom application and validate that it is working before executing the actual Content Manager product migration on a production system.

In this section, we provide an overview for the porting process. To help you better understand the changes in APIs, we also briefly outline what’s new in Content Manager V8.3 APIs (and earlier V8 versions).

17.1.1 Porting process

To migrate a customer application, we recommend applying this porting process:

1. First, understand the Content Manager V8.3 system, particularly the new data model (refer to Chapter 3, “Data modeling” on page 29 for details). See how the new data model can work with your current application requirements and specifications, using flow charts or concept diagrams to illustrate correlations.

2. Analyze and understand your current application; identify the areas that can be improved or enhanced, in order to:
   – Make your application more efficient and run faster
   – Add more functionality to your application
   – Match original specifications that were not available with earlier releases
   – Make it possible to implement new requirements more easily

3. Read all sections of this chapter carefully and estimate the application migration effort. Understand all the available APIs. You need to know what APIs your application uses and to what extent it does. This gives you an idea of the tasks involved as well as the size of the individual tasks.

4. Set up a test and development system. We suggest that you use a separate test system to do the application migration. The test system should include real data copied from the production system.

5. Port the application. Concentrate on the sections of your APIs in order of importance, or in a hierarchy that you think will benefit you. You may address the changes to your application in the following order:
   a. Setting up administrative tasks, such as logon, logoff, and user privileges
   b. Defining the servers
   c. Working with data and how it is being created, retrieved, updated, and deleted
   d. Working with items and objects and how to take advantage of links and attributes
   e. Performing other overhead tasks
Notice that these changes not only deal with making the application run with the new release of the product, but they also deal with the improvements now possible with the new version.

6. After the application migration is completed, perform an extensive functional verification and integration test. Address any issues as they arise.

The preceding process is a high-level, recommended migrating procedure. It is not intended as a project plan; however, it does reflect how application migration fits into the overall Content Manager migration process and it emphasizes the major tasks which you need to work on. Any concrete planning for an actual application migration project should be performed by an expert.

**Note:** The source material of this chapter comes from Chapter 9 of the following redbook:

*Content Manager Version 8.1 Migration Guide for Multiplatforms*, SG24-6877

If time allows, we recommend reading Chapter 9, Chapter 10, and the corresponding Appendixes of the redbook for more information.

### 17.1.2 What's new in V8.3 APIs from V7

In this section, we consolidate and outline all the new APIs introduced in V8.1, V8.2, to V8.3 from V7.

The new features and components include the following capabilities:

- **Storing content in user-specified collection:** New APIs have been introduced allowing you to specify the storage location for a Resource Item of Document Part. This overrides the administrative setting for the default storage location.

  See

  `DKStorageManageInfoICM::setStorageLocation(java.lang.StringRMName, java.lang.StringCollectionName)`

  **Usage:**

  ```java
  DKStorageManageInfoICM storageInfo = (DKStorageManageInfoICM) lob.getExtension("DKStorageManageInfoICM");
  storageInfo.setStoreLocation("RM1","Coll1");
  ....
  lob.add();
  ```

- **Query for content that is checked out:** The API now allows you to query for content that has been checked out by any user. See 7.3.14, “Query on checked-out items” on page 182 for more details.
- **Content Manager C++ API global cache:** As of Content Manager 8.3 Fix Pack 2, the Content Manager C++ API uses a global cache, reducing the communication required between application and Library Server. For more information, see Chapter 20, “Performance tuning” on page 543, for more details.

- **Content Manager C++ API datastore pool:** As of Content Manager 8.3 Fix Pack 2, the Content Manager C++ API provides a datastore pool. This can significantly improve the performance of applications written using the Content Manager C++ API. For more information, see Chapter 20, “Performance tuning” on page 543, for more details.

- **Content Manager C++ API database connection pool:** As of Content Manager 8.3 Fix Pack 2, the Content Manager C++ API provides a database connection pool. This can significantly improve the performance of applications written using the Content Manager C++ API. The database connection pool works similarly to the datastore pool, but the connection pool works transparently at the lower level of actual database connections to the Library Server database. For more information, see Chapter 20, “Performance tuning” on page 543, for more details.

- **Federated folders (Java only):** Information Integrator for Content Version 8.2 now provides special federated entities that can hold *federated folders*. These federated folders can store the combined results from a federated query, such as a document from Content Manager and a related document from OnDemand. You can then send the results directly into a workflow.

- **Microsoft Visual Studio .NET support:** APIs now support Microsoft Visual Studio .NET.

- **XML import capabilities:** You can now use XML to import and export content into Content Manager through DDOs and XDOs (using Java APIs).

- **Additional connectors for relational databases:** These include relational database connectors for DB2 UDB, DB2 DataJoiner, DB2 Data Warehouse Manager Information Catalog Manager, and other databases through JDBC or ODBC drivers.

- **Workflow capabilities:** By using Information Integrator for Content workflow feature, you can define and run the workflow process of a work group, department, or enterprise.

- **Federated level access control:** You can control access to workflow processes through the use of privilege sets and access control lists. Additional access control to data can be managed by the access control features of each content server.
Additional support for Content Manager: This includes the following capabilities:
- List, add, retrieve, update, and delete of content class
- Asynchronous retrieval of object content

17.2 Application porting scenarios overview

There are many ways to programmatically port or migrate an application using the APIs and application development toolkits available with Content Manager V8.3. This section is intended to give you an idea of the APIs that are available and where they exist in the new Content Manager V8.3 data model.

Figure 17-1 provides an architectural overview of the software components and the APIs present in Content Manager V7:

- The Thin Client or Client Layer
- The EIP Unified Portal or Federated Layer
- The OO API or DL Connector Layer
- The C API or Folder Manager Layer
- The Communication Layer
Figure 17-2 gives an architectural overview of the software components and APIs present in the Content Manager V8.3. In comparison to the V7, it is clear that the Folder Manager and Service layer is gone, as well as the Text and Image Search Servers. These have been pushed further down the stack and their functionality is being provided by the underlying DB2 database Stored Procedures and Extenders. The OO API layer now comprises the ICM Connectors. The control of the Object Server is being handled via HTTP using the Websphere Application Server.
In the remaining sections of this chapter we give a brief description of each API as it is defined in Version 8. For more information, refer to the appropriate product documentation and the Information Center that comes with Content Manager. The migration or porting of the APIs is discussed here.

### 17.3 Information Integrator for Content Java beans

The Information Integrator for Content JavaBeans are designed to ease development of end-user applications. The beans follow JavaBeans conventions, with default constructors, properties and events. Also, the beans include associated BeanInfo classes, which aid their use in visual builder environments. There are visual and non-visual JavaBeans for use in Java applications.

- **Nonvisual beans**: These beans are useful in building Web applications and other Java applications.
- **Visual beans**: These beans provide user interface “panels” of an application and can be used to quickly build Swing-based Java client applications.
- **Java viewer toolkit**: These classes are used by the nonvisual and visual beans to provide document conversion, document rendering, and graphical annotations editing. They can also be used independently of the beans for standalone viewer applications/applets.

**Porting**

Applications using the Java beans layer are fairly easy to port from Content Manager V7 to Content Manager V8.3. The largest part should work on the new version without changes.

Content Manager V8.3 offers new functionality for which the Java beans have been extended. Analyze your application to see if it benefits from these enhancements.

The two major changes to previous functionality are:

- **Adding parts**: In Content Manager V8, the concept of a part type has been added. Since the concept is new, you must adapt your code to explicitly set the part type on each part that you add.

- **Expressing CM queries**: In Content Manager V8, the query syntax has changed from the former proprietary syntax to an XML-based query language that conforms to XQuery Path Expressions (XQPE), a subset of the W3C XML Query working draft. This is a powerful new function; but it means that the syntax of all your queries must be checked and adapted. Note that this applies only to the query string, not to the method calls.

For detailed information on the terminology of this section and how to actually perform the suggested changes, please refer to the following redbook:

- *Content Manager for Multiplatforms Version 8.1 Migration Guide, SG24-6877*

### 17.4 Information Integrator for Content Federated Connector

The Information Integrator for Content (EIP) Federated Connector enables searches across many different data sources such as content servers. This process is called *federated* searching. For a complete list of servers, refer to the *Content Manager for Multiplatforms Version 8.1 Migration Guide, SG24-6877*.

The mapping of content server data structures to the federated document model is subject to some content server specific restrictions. For example, a DB2 database does not have folders or parts; a federated document maps to a row in a DB2 table or other relational database.
Its primary use is to facilitate the integration of several content servers in order to enable simultaneous federated searches across them. The EIP Federated Connector can only be accessed through an object-oriented API, which is available in Java and in C++ as well as in OLE Automation interface.

**Porting**
The interface to the EIP Federated Connector has remained unchanged from Version 7, so there is no porting effort. If you are using C++, you must recompile the code that access the Federated Connector classes.

For detailed information on the terminology of this section and how to actually perform the suggested changes, please refer to the *EIP Application Programming Guide* and *Content Manager for Multiplatforms Version 8.1 Migration Guide*, SG24-6877.

### 17.5 Information Integrator for Content DL Connector

The Information Integrator for Content (EIP) DL Connector provides an object-oriented interface to the Content Manager V6 and V7. The new Content Manager Version 8 ICM Connector is an extension of the Content Manager V7 DL connector and includes a variety of functional enhancements.

The ICM Datastore includes all of the supporting classes that are required to connect to a Content Manager Version 8 server. It provides the concepts and enables operations on hierarchical items, versioning, links, references, and query and cursor support, including metadata manipulations. You cannot use applications developed for Version 7 with the ICM Datastore. You must rewrite applications with the new object-oriented APIs to exploit the new features of Content Manager Version 8.

**Porting**
The EIP DL Connector continues to exist in Information Integrator for Content V8, so porting is not necessarily required. If you decide not to port, be aware that this means you will continue to use Content Manager V7. If you migrated your current system to Content Manager V8, you must change the applications to use the EIP ICM Connector instead of the EIP DL Connector. Such a change involves a porting effort.

Refer to Appendix E, “API migration tables for Content Manager” on page 669 for a reference of DL Connector method calls and how they map to the new ICM connector method calls. Refer to 17.1.1, “Porting process” on page 452 for suggestions on planning the porting process.
17.6 CM Folder Manager and Library Client API

The Folder Manager (FM) API is the lowest level at which a Content Manager V7 system is programmable. The FM API consists of two layers. The Folder Manager layer uses functions of the Library Client API.

The FM API is primarily used when the following situations are applicable:

- The application is using the C programming language.
- The data model matches the FM logical data model.
- The function and speed of a lower level API are required.

If you are not familiar with these concepts, please refer to the IBM Content Manager Version 7.1 Application Programming Guide and the IBM Content Manager Version 7.1 System Administration Guide.

**Porting**

Since Content Manager V8 no longer supports FM API and the C based interfaces, porting an FM API based application means completely rewriting your application to support Content Manager V8.

Your application must be rewritten to use the object oriented API of the EIP ICM Connector. This also involves changing the programming language from C to Java or to C++.

Refer to Appendix E, “API migration tables for Content Manager” on page 669 for a reference of FM API function calls and how they map to the new ICM Connector method calls.

17.7 Text Search Engine (TSE)

Text Search is another feature of EIP V7 which has been removed from Information Integrator for Content V8 and replaced by DB2 Net Search Extender (NSE). The Text Search Engine (TSE) is used to automatically index, search, and retrieve documents stored in Content Manager V7. Users can locate documents by searching for words or phrases. TSE is a feature of EIP V7 which can be accessed through Java and C++ APIs.

**Porting**

Information Integrator for Content V8.3 still includes the TSE feature; however, this feature only works with documents stored in Content Manager V7. Using a Content Manager V7 back-end server, you may continue to use the corresponding Text Search Engine and still take advantage of the other features of Information Integrator for Content V8. This does not require porting. If you use
or will use a Content Manager V8 back-end for storing text-searchable documents, then you must port your application to use the new text search functionality, using Net Search Extender.

17.8 Image Search Engine (QBIC)

The Image Search Engine is a feature of EIP V7. It is based on IBM QBIC® technology and allows the indexing and searching for images by certain visual properties, such as color, shape, or histogram.

The Image Search Engine is primarily used in the following cases:
- The application uses the object-oriented APIs in Java or C++.
- The application requires the capability to search for stored parts based on their visual properties.

**Porting**

Information Integrator for Content V8 still includes the Image Search Engine feature; however, this feature is indeed the Image Search Engine from Version 7, which only works with parts stored in Content Manager V7 back-end. Using a Content Manager V7 back-end server, you may continue to use the corresponding Image Search Engine and still take advantage of the other features of Information Integrator for Content V8. This does not require porting. Since Content Manager V8 no longer supports image searching, porting is not possible.

**Note:** If your application must have image search capability at all costs, then we recommend using the IBM DB2 AVI Extender, which can perform image searching; but, it is not integrated with Content Manager. It may also be feasible to integrate external software into your code which performs the image indexing, for example, GIFT, the GNU Image Finding Tool. Be aware that either approach incurs a significant development effort.

17.9 Client for Windows Automation Interface

The Content Manager Client for Windows has provided the ability to call functions in the client from external programs since Version 2. This can be used to either invoke client GUI functions or to retrieve and store Library Server data via programs using the functions provided by the client. The interface is known by a number of different names (OLE, ActiveX or COM); but the official name used is the Client For Windows Automation Interface and is based on the OLE 2.0 Automation architecture.
The interface is object oriented and it enables very easy integration with programming languages supporting OLE such as C++ and MS VisualBasic (VB). In addition, a number of scripting languages, such as Lotus Script used in Lotus Notes, are also supported. For detailed information on the Automation Interface, please see the manual *IBM Content Manager Client for Windows: Client for Windows Programming Reference*, SC27-1337.

The Automation Interface is used in the following cases:

- Quick integration with Content Manager from word processing programs such as MS Word and Lotus Word Pro by calling library functions directly from scripts.
- Extend C++ and VB programs with Content Manager library functions without writing the necessary GUI, but invoke the Clients search and view functions.
- Store and retrieve functions from C++ and VB programs without using the GUI.

**Porting**

The basic object model of the Automation Interface has not changed in Content Manager V8 compared to the model used in Content Manager V6 or V7. In order to utilize the new Content Manager data model and access to the improved functionality in the new version, some methods and properties have been added to the Automation Interface; more importantly, they are:

- The Client for Windows main program is now called *ICMClient.exe* (as opposed to *VIC.exe* in Version 7). You must use this new name when creating instances of the automation interface. The old and new version of the client can co-exist on the same end-user machine. This makes it easier to implement and test your migration efforts. It also enables you to have both versions running simultaneously in a production scenario if you need to access both versions of the library at the same time.
- A number of properties and methods which are implemented in Version 7 have been removed.
- Many methods and properties have been renamed to match the naming of the new Content Manager data model. For example, *Index Class* is now termed *ItemType*. This leads to the renaming of the method call from *ClassArray* to *ItemTypeArray*.
- The data type accepted or returned by some properties and methods may have changed.

Because of the preceding listed changes, we evaluate the porting effort to be from a medium to a major effort because you cannot just perform a “global change” in your program to use the new naming structure; you have to review all the method and property references to ensure correct naming and the data types passed and received in calls.
17.10 Information Mining based applications

Information mining in Information Integrator for Content V8 is based on a totally new engine; it is incompatible with the Information Mining delivered in Content Manager EIP V7. For customers that have EIP V7 Information Mining installed, a reload of the data is required and retraining of the Catalogue engine needs to be performed.

*Porting*

Since there is no migration tool available for this, you need to rewrite the application to unload and reload the data.

17.11 OLE API based applications

In order to integrate LOB applications with Content Manager functions with minimal effort, most customers use the OLE automation APIs delivered with the Content Manager Windows client to manipulate the Windows Client functionality from within their LOB application. This interface can be called from programming environments such as Visual Basic, Visual C++®, and Power Builder.

This capability was included in Content Manager V8 as well; but customers that use this must be aware of the following modifications which need to be done to an application using this technique with Content Manager V7.

Content Manager V7 delivered the OLE APIs, which were built on top of the Folder Manager APIs. The parameters passed to the OLE APIs in Version 7 were dictated by the Folder Manager APIs parameter requirement (since they had to be mapped). The Content Manager V8 Client uses the C++ OO APIs, which in certain circumstances, require different parameters to be passed. The customer must verify that the OLE methods have the appropriate parameters required by Content Manager V8 Client OLE APIs.

Content Manager V7 also delivered sample Visual Basic routines that included several OLE APIs underneath to facilitate the coding. This is no longer available in Content Manager V8 Client. The old sample Visual Basic code can be upgraded to work with the new Content Manager V8 OLE APIs.

17.12 APIs other than FM APIs not carried into CM V8

There are APIs in EIP V7 and in Information Integrator for Content and Content Manager V8 connectors which customers may have used to write applications
with, but that they should re-evaluate as they move into the Content Manager V8 platform and into the future. One example of these APIs includes the “Rights Management APIs”, which allowed users to place watermarks into images for protection of their information. Although these APIs continue to be delivered in the different connectors, they are not deemed strategic. You must re-evaluate their use as this technology evolves. Table 17-1 list some of these APIs.

<table>
<thead>
<tr>
<th>APIs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rnpwigi</td>
<td>Initialize the generic marking environment</td>
</tr>
<tr>
<td>fnpwigt</td>
<td>Terminate the generic marking environment</td>
</tr>
<tr>
<td>fnpwigo</td>
<td>Open a marking handle to the environment for a specific image</td>
</tr>
<tr>
<td>fnpwigr</td>
<td>Read marked data to a buffer from a marking handle</td>
</tr>
<tr>
<td>fnpwigc</td>
<td>Close the handle to a marking environment for a specific image</td>
</tr>
<tr>
<td>fnpwihe</td>
<td>Return the error code stored in a marking handle</td>
</tr>
<tr>
<td>fnpwigc</td>
<td>Extract the generic API part of the error code (higher two bytes)</td>
</tr>
<tr>
<td>fnpwipe</td>
<td>Extract the plug-in part of the error code (lower two bytes)</td>
</tr>
</tbody>
</table>

### 17.13 Programming tips

This section provides programming tips that you can use when migrating your applications. Also refer to “Application development overview” on page 131 for general application development overview.

#### 17.13.1 Packaging for the Java environment

The Information Integrator for Content APIs are contained in four packages as part of com.ibm.mm.sdk: server, client, common, and cs, as follows:

- **server** (com.ibm.mm.sdk.server)
  
  Access and manipulate content server information.

- **client** (com.ibm.mm.sdk.client)
  
  Communicate with the server package using Remote Method Invocation (RMI).

- **common** (com.ibm.mm.sdk.common)
  
  Common classes for the server package, client package, and cs package.
Chapter 17. Application migration

- cs (com.ibm.mm.sdk.cs)

  Connect the client or server dynamically

Your application must use the common with either the server package for local applications, or the client package for applications that access the remote server, or the cs package.

**Tip:** Do not import client and server packages in the same program. If you are developing a client application, import the client package. Otherwise, import the server package. If you do not know where the content resides, then use the cs package (with the server or client packages). Importing multiple packages can result in compile errors.

### 17.13.2 Programming using Content Manager V8

When programming for Content Manager V8, we recommend the following programming tips and considerations:

- Java programming tips
- C++ programming tips
- Considerations when using the non-visual beans
- Singleton in the Beans
- Threading considerations in the Beans

**Java programming tips**

For Content Manager V8 and later, an XDO is a dkResource object. You use DKPidICM to represent the PID of the resource object. For earlier Content Manager, Content Manager for AS/400®, and IP 390, you identify an XDO by the combination of item ID, part ID and the RepType. For RDB, the key to identify an XDO is combination of table, column and data predicate string. To handle a stand-alone XDO, you provide the item ID and part ID. The RepType is optional since the system provides a default value for it.

**C++ programming tips**

For Content Manager, VI400 and IP390, you identify an XDO by the combination of item ID, part ID, and RepType. For Relational Databases, the combination of table name, column name and datapredicate is the key to identify an XDO. For a standalone XDO, you must provide the item ID and part ID. RepType is optional, because the system provides a default value (FRN$NULL). For the add function, you must provide a part ID. You can retrieve the part ID value after add if you want to do some other operation with that object later.
Important: When adding a part for the search manager to index on a Content Manager content server, you must have a valid part ID and cannot set the part ID to 0.

Considerations when using the non-visual beans
You can use the non-visual beans to enable general-purpose applications with the functionality required to access content management repositories supported by Information Integrator for Content.

Singletons in the Beans
CMBConnection has methods to obtain access to instances of the other session-wide EIP Beans. When the session-wide beans such as CMBSchemaManagement and CMBDataManagement are obtained in this way, they are already wired to the CMBConnection bean (from which they are obtained) to be informed of a connection or a disconnection, and to share trace and exception event handlers.

Only a single instance of each of the other session-wide beans is created. If these methods are called repeatedly, the same instance is returned (singleton design pattern). If session-wide beans are created in the application, and not by the CMBConnection bean, they must be wired to a CMBConnection bean to be used.

Threading considerations in the Beans
A single instance of the CMBConnection bean can only be used on a single thread at any point in time. This restriction extends to all other beans that are associated to a CMBConnection bean (through the connection property of the associated bean). That means that you must create separate connections for each thread. Alternatively, multiple threads can obtain and free connections using the CMBConnectionPool bean. Therefore, each thread should obtain, use, and free a connection.

All the session-wide beans have affinities to an instance of the CMBConnection from which they were retrieved or with which they have been associated after creation. This implies that an instance of the session-wide beans such as CMBSchemaManagement can only be used by a single thread at any given time. If the session-wide bean instance is used by multiple threads, you must perform explicit synchronization in your application to ensure that only a single thread is actively using the session-wide bean instance at any given time. All session-wide beans also listen to connection reply events generated by the CMBConnection beans. This allows them to recognize that the underlying content repository with which the CMBConnection bean instance is associated has changed, so that the beans can take appropriate action.
Unlike the CMBConnection, the CMBConnectionPool bean is designed for multithreaded use. Multiple threads can simultaneously call the methods related to obtaining and freeing connection objects. Any connection obtained from the pool is an instance of CMBConnection and is restricted to single-thread access. Any connection obtained from the connection pool bean should be returned to the pool as soon as possible after its use, so that it may be made available to other threads that might be requesting connections from the pool.

### 17.13.3 Working with the Content Manager samples

Content Manager provides a comprehensive set of code samples to help you complete key Content Manager tasks. The samples are a great source of API education because they provide reference information, programming guidance, API usage examples, and tools. Samples can be viewed using *Online application Programming Reference*, in the product Information Center. Additionally, the samples are located in the following directories:

- C:\Program Files\IBM\db2cmv8\samples\java\icm
- C:\Program Files\IBM\db2cmv8\samples\cpp\icm

**Note:** You must have selected the Samples and Tools component during Information Integrator for Content installation in order to have the samples in the directory.

To get the most out of the samples, be sure to read the Samples Readme. It contains a complete reference index to help you quickly find the sample that contains the concept, or topic, that you are looking for. Every sample is thoroughly documented and provides in-depth conceptual information and an explanation of each task step. Additional information contained in each sample includes:

- Detailed header information explaining the concepts shown in the sample.
- A description of the sample file including prerequisite information and command line usage.
- Fully commented code that you can easily cut, customize, and use in your applications.
- Utility functions that you can use when developing your applications.

Refer to 6.1.5, “Working with sample code” on page 139 for more information.
Part 5

Maintenance

In this part of the book, we discuss maintenance activities. Once a Content Manager system is implemented or migrated, it is important to maintain the system. We cover maintenance issues, including regular maintenance procedures, performance tuning, and troubleshooting hints and tips for a production Content Manager system.
Maintenance

In this chapter, we describe the maintenance tasks necessary to keep a Content Manager system up and running. This includes a discussion on what data needs to be backed up in order to restore an entire system, as well as information about monitoring tasks that should be performed on a regular basis.
18.1 Maintenance tasks overview

Once a Content Manager system has been designed, installed, and configured, it is still important to perform maintenance tasks regularly.

By maintaining a Content Manager system properly and in a timely manner, you can get the best possible performance from it over time, and you will potentially avoid problems that can manifest themselves as system endangering situations.

Much of this chapter is an extract from IBM Content Manager System Administration Guide v8.2, SC27-1335; the publication can be used in conjunction with this redbook to provide further details.

The regular maintenance tasks include the following activities:

► “Optimizing server databases” on page 472
► “Monitoring LBOSDATA directory size” on page 477
► “Managing staging directory space” on page 480
► “Removing entries from the events table” on page 483
► “Removing log files” on page 484
► “Managing Resource Manager utilities and services” on page 485
► “Replacing or repartitioning a hard disk” on page 499
► “Backup” on page 502

18.2 Optimizing server databases

Database statistics should be updated periodically, daily or weekly depending on the ingest load, on the Library Server and Resource Manager databases in order to maintain good performance. This should also be the first step whenever it appears that there are problems associated with Library Server performance, such as slower logons, searches, or indexing.

**RUNSTATS/REBIND database tables**

Keeping DB2 statistics up to date on the tables and data helps the optimizer to choose the best execution access plans for SQL statements to maximize performance. We recommend that you run **RUNSTATS** and **REBIND** commands on a regular basis as part of regular database maintenance. Recalculating table statistics is **critical** to improving database performance and should be done regularly. If a recalculation of table statistics has not been done recently, then this should be the first step in diagnosing DB2 performance issues.

You can find instructions about how to run these commands in the DB2 Command Reference (click **Start** → **Programs** → **IBM DB2** → **Information** → **DB2 Information Center** and type **runstats** or **rebinds** in the search field).
Use the DB2 Command Reference, SC09-4828 and the following instructions to execute these commands:

1. Open a DB2 Command Window by clicking **Start** → **Programs** → **IBM DB2** → **Command Line Tools** → **Command Window**. If you are using a UNIX machine to perform these tasks, once the db2profile script has been run (if it is not in root’s .profile), the commands can be typed directly onto the UNIX command line. If you are not already connected to the database, connect to the database by entering:

   ```
   db2 connect to <db name> user <user ID> using <password>
   ```

   Where:
   - `<db name>` is the name of the database.
   - `<user ID>` is user ID with administration rights on the database.
   - `<password>` is the valid password of the user ID.

2. Run **RUNSTAT** as follows:

   ```
   db2 runstats on table <table_name> with distribution and detailed indexes all
   ```

   This should be done for each table in the database, for both the Library Server and Resource Manager databases.

   For example, for SYSINDEXES table, use:

   ```
   db2 runstats on table sysibm.sysindexes
   ```

3. The DB2 system **DB2RBIND** command should be executed after calculating table statistics to rebind all the packages in the database. This is a system command and not a command-line statement. This means it does not have to be prefixed with `db2` like the **RUNSTATS** command does.

   Run **DB2RBIND** as follows:

   ```
   db2rbind <database_name> -l <logfile name> all /u <userid> /p <password>
   ```

   For example

   ```
   db2rbind icmnlsdb -l bind.log all /u icmadmin /p password
   db2rbind rmdb -l bind.log all /u rmadmin /p password
   ```

4. Check your log file to see the results. Another way that you can check the success of a rebind is by using the DB2 Control Center:

   a. Open the Control Center by clicking **Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**.

   b. Go to the database against which you ran **DB2RBIND**.

   c. In the database, go to **Application objects** → **Packages**.
Check the columns, Last bind date, and Last bind time. The date and time indicate when you last had DB2 rebind all the packages.

**REORGCHK**

A table can become fragmented after many updates, causing performance to deteriorate. Queries take longer because index entries in the Library Server and Resource Manager are no longer synchronized with the actual data in the database tables.

You can synchronize the data in the index with the database tables by running the REORGCHK command in DB2. The REORGCHK command gathers and compares both the index and the table statistics and recommends tables to reorganize. Following the recommendation, use REORG command to reorganize the necessary tables.

When you reorganize tables, you remove empty spaces and arrange table data for efficient access. Reorganizing tables takes a lot more time than simply checking (REORGCHK) which tables may require reorganization. Do not reorganize tables when you expect a lot of server activity, because performance may be impacted. DB2 locks any data in a table that is currently being reorganized.

If you update tables often, then you want to reorganize periodically, for example, once a month. If you do not manage the DB2 database tables, you need to work with the DB2 administrator for access or to coordinate when to run REORGCHK and when to reorganize tables.

You can find instructions about how to update database tables in the DB2 Command Reference (click Start → Programs → IBM DB2 → Information → DB2 Information Center and type reorgchk in the search field). Use the DB2 Command Reference, SC09-4828 and the following instructions to check and update database tables:

1. Open a DB2 Command Window by clicking Start → Programs → IBM DB2 → Command Line Tools → Command Window. If you are using a UNIX machine to perform these tasks, once the db2profile script has been run (if it is not in root's .profile), the commands can be typed directly onto the UNIX command line. If you are not already connected to the database, connect to the database by entering:

```
   db2 connect to <db name> user <user ID> using <password>
```

Where:

- `<db name>` is the name of the database.
- `<user ID>` is user ID with administration rights on the database.
- `<password>` is the valid password of the user ID.
2. Run REORGCHK as follows:

```
db2 reorgchk update statistics on table all > out.txt
```

Where

out.txt is the log file where the output is generated into.

Note that when you run REORGCHK, we recommend storing the results in a file, due to the large amount of output generated. This file, known as a log file, contains the statistics you need to use to determine whether to reorganize a particular table. In our scenario, we pipe the results to out.txt. You can use any name for the log file.

If you have a general idea of what tables usually need to be reorganized, you can perform REORGCHK on only these tables.

3. Check the REORG column in your log file. DB2 displays 1 to 3 asterisks (***) in the REORG column when it detects a table to reorganize. The number of asterisks determine the urgency of reorganizing the table.

The first two columns are the schema name and table name. You use these two names to reorganize tables. For example, a schema name can be icmadmin or sysibm and a table name can be ICMSTNLSKEYWORDS or SYSINDEXES.

4. To reorganize a particular table, use:

```
db2 reorg table <table name>
```

Where

<table name> is the name of the table you want to reorganize.

For example, to reorganize SYSINDEXES table, we use:

```
db2 reorg table sysibm.sysindexes
```

5. If you use REORG on any table, it is also a good idea to use the RUNSTATS command to update the table statistics again:

```
db2 runstats on table <table name>
```

Where

<table name> is the name of the table you want to update its statistics.

For example, for SYSINDEXES table, we use:

```
db2 runstats on table sysibm.sysindexes
```

6. When you finish reorganizing database tables, you need to rebind all packages within the database using the DB2RBIND command. This is to allow new access plans to be generated. You do not need to be connected to the database for this step. In the DB2 Command Window enter:

```
db2rbind <db name> /l report.txt
```
Where:

- `<db name>` is the name of the database.
- `report.txt` is the name of the log file that contains any errors that results from the package revalidation procedure.

**Important:** You need a user ID and password if you plan to update a schema that does not belong to you. Also, the user ID and password must have DB2 administrative authority to complete this task.

This command uses the rebind API (`SQLARBND`) to attempt the revalidation of all packages in a database. If the rebind of any of the packages encounters a deadlock or a lock time out, the rebind of all the packages will be rolled back.

7. Check your log file to see the results. Another way that you can check the success of a rebind is by using the DB2 Control Center:
   a. Open the Control Center by clicking `Start` → `Programs` → `IBM DB2` → `General Administration Tools` → `Control Center`.
   b. Go to the database against which you ran `DB2RBIND`.
   c. In the database, go to `Application objects` → `Packages`.
   d. Check the columns, `Last bind date`, and `Last bind time`. The date and time indicate when you last had DB2 rebind all the packages (see Figure 18-1).
Figure 18-1  Checking the success of a rebind using the DB2 Control Center

For more information about RUNSTATS, REBIND, REORGCHK, and other DB2 commands, see the DB2 Command Reference, SC09-4828. For a more detailed understanding of reorganizing and rebinding DB2 database tables, see the DB2 System Administration Guide, SC09-4820.

18.3 Monitoring LBOSDATA directory size

The LBOSDATA directory is an area of local disk that a Resource Manager controls, and is used to store objects.

When using fixed disk attached to the Resource Managers for object storage, it is very important to make sure that there is enough free space remaining for Content Manager to write objects to. If Content Manager runs out of space to write objects to, any new requests to store objects will fail.
Even when Tivoli Storage Manager (TSM) is used for the long term storage of objects, Content Manager may be configured to keep objects locally, and only migrate to TSM after a period of time (for example, 30 days). The migration of objects to TSM is triggered by the length of time the objects have resided in the first storage class, assuming there are only two storage classes and TSM is the second one.

In this instance, it is possible that the local fixed volume that the LBOSDATA directory resides on, may become completely full if new objects are being added to it faster than they are being migrated to TSM by the Content Manager migrator process. This may occur during peak periods of object loading into Content Manager, such as around the end of a financial year for an accounting company that scans documents into Content Manager for reference purposes. In worst case scenario, the process of migrating from LBOSDATA directory to TSM may not even be running.

**Important:** It is important to remember to have the Content Manager migrator process running at all times, even if you do not migrate objects between storage classes. That is because it is used to physically delete objects from where the Resource Manager has stored them. When an end user deletes an object from the standard client, only the row from the Library Server database is deleted immediately (for performance reasons), the entry in the Resource Manager database and the object itself remain. The migrator must be run to reclaim the physical storage space.

During peak activity times, it is even more important to monitor the amount of free space remaining within the local fixed disk that the LBOSDATA directory resides on. Of course, these peak periods of activity should have been taken into account when designing and sizing the system; nevertheless, monitoring the directories is good practice. Operating system tools should be used to monitor the current space occupied by the local objects, and the amount of space remaining on a physical or logical volume.

It is possible to see how many MBs of storage remain on a particular file system volume (see Figure 18-2 on page 479), through the Content Manager System Administration Client; however, this value does not get updated dynamically. If space is being used on a file system volume while you are logged onto the System Administration Client, the number of MBs free on the file system volume will not change. To see the change in volume free space, you need to log off and log back onto the System Administration Client. Note that this is not an entirely accurate way to monitor free space.

When you create a file system volume for Content Manager to use, such as the e: drive on a Windows machine, Content Manager takes the remaining free
space available on this physical or logical volume, as the amount of free space currently available for objects. In other words, when you create a file system volume within Content Manager, you cannot assign only a certain percentage of it to be used.

In the same way, Content Manager does not reserve space for its objects on a file system volume. It is very important to make sure that no other applications use the same volume to store dynamic data. If this occurs, the amount of free space available to Content Manager to write objects can unexpectedly be changed.

Figure 18-2 shows the window that is used to define a new file system volume to a Resource Manager.

---

**Figure 18-2** Defining a new file system volume
Should a file system volume become full, it is possible to define a new volume, assuming you have the physical space available, and then add this new volume to the existing storage group, in order to provide further space for Content Manager to store objects to.

When you define a file system volume, a threshold percentage can be entered (see Figure 18-2). This value is used as a limit at which point Content Manager will attempt to migrate objects to the next storage class, if one exists, and if the migrator process is running. The default value for the threshold is 95%. This threshold limit should never be reached in the normal course of events, and the threshold limit mechanism should not be relied upon as the default way of monitoring and dealing with overly full volumes. In most circumstances, the default value of 95% will be fine to use in a production system.

A more effective way to prevent Content Manager running out of space is to create overflow volumes which are volumes that can be used by any storage groups, when all other storage systems, such as a file system volume, within a storage group are full. To create an overflow volume, select **Overflow** under the Assignment section of the New File System Volume window (see Figure 18-2 on page 479). You can define as many overflow volumes as you desire.

It is also important to monitor the space remaining in the TSM volumes used by Content Manager, as well as the space remaining in your TSM database, and log volumes. You should contact your TSM administrator in order to perform these functions.

### 18.4 Managing staging directory space

The staging area is used as a temporary storage area for objects retrieved from TSM storage and as the location to store objects when the LAN cache is enabled for a Resource Manager. Using the staging area enables faster response time for subsequent retrievals of the same objects.

The System Administration Client allows users to manage the staging directory to get the most benefits from LAN caching and also from TSM object retrieval caching. Staging directory management tasks include:

- Setting automatic cache purge specifications: A purge removes the oldest, least frequently used objects from the staging directory.
- Defining subdirectories to hold cached objects: Storing cached objects in subdirectories can improve system retrieval time because the system can target the search without looking through individual objects stored in the staging directory.
Defining the size of the staging directory: Depending on the size and volume of cached objects, you may need to modify the original parameters defined for the staging directory.

Figure 18-3 shows the Staging Area properties window which is accessed through the Content Manager System Administration Client. Right-click on the Resource Manager database, and then select Staging Area.

![Staging Area properties window](image)

Defining the maximum size of the cached object. Note that the system will not cache objects that exceed the maximum size; however, if you decrease the maximum size and objects that were stored earlier exceed the new maximum size, the system will retain these existing objects.

### 18.4.1 Purger process

The purger process is used to maintain the size of the staging area. When the staging area size reaches a preset upper limit, the purger will begin to remove files until it reaches a preset lower limit. Using Figure 18-3 as our example, this means our staging area is 199 MB in size, and purging will commence when this 199 MB area is 80% (159.2MB) full, providing the purger process is started. Once the staging area reaches 159.2 MB full in size, the purger will start randomly deleting files until the staging area reaches 60% of 199 MB (119.4MB).
All of the staging area values are configurable. For example, if you want to completely clear the staging area, you can set the start purge size to 1% of the maximum staging area, and the stop purge size to 0% of the maximum staging area size. Figure 18-4 shows this configuration.

![Staging Area - RMDB](image)

*Figure 18-4  Configuration to clear staging area completely*

With the configuration set in Figure 18-4, you should be able to clear the entire staging area assuming that the staging area is at least 1% of 199 MB full at the time. If the staging area is below 1% full at the time, you need to reduce the size of the staging area down from 199 MB to a size where 1% of the staging area maximum size was smaller than the currently occupied space within the staging area.

The staging area maximum size and purge rates are monitored periodically, not constantly. For this reason, you may need to wait up to five minutes, the default setting, before changes you have made to the staging area come into effect. The cycle time for this checking is configured via the Resource Manager configuration window. To open this window, go to the Content Manager System Administration Client, open a Resource Manager and select Configurations. Then select the Resource Manager configuration that you are currently using, the default is IBMCONFIG, and select the tab labelled Cycles (see Figure 18-5).
Figure 18-5  Resource Manager cycles

The threshold cycle sets the amount of time that elapses before the staging area size is updated. Figure 18-5 displays the defaults for a Resource Manager. The other cycles refer to amount of time that elapses before the various Resource Manager utilities check to see if they have any work to do.

The settings for the staging area and cycle times that are best suited to your environment may differ from the default settings. For example, if your system produces instances when the staging area is heavily used, you may need to adjust the cycle time so that the purger checks the staging area more regularly to see if it has any work to do.

18.5 Removing entries from the events table

When you use the Content Management System Administration Client, the Library Server records item and document routing related functions in the events table, icmsts.sysadmevents or icmstitemevents.
The events table grows with each logged event. To reduce the size of the events table, you can remove the expired and unused events from the table. The EventCode column in the events table indicates the classification of events as the following values, shown in Table 18-1.

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 200</td>
<td>System administration function event codes</td>
</tr>
<tr>
<td>200 - 900</td>
<td>Item, document routing, and resource management function event codes</td>
</tr>
<tr>
<td>1000 +</td>
<td>Application event codes</td>
</tr>
</tbody>
</table>

You can delete events from the events table by performing either of these following tasks:

- To delete an event for a system administration function from a Library Server, connect to your database and use the following SQL command:
  
  ```
  delete from ICMSTSYSADMEVENTS where eventcode <=200 and Created < 2002-01-01-12.00.00.000000
  ```

- To delete an event for an item function from a Library Server, connect to your database and use the following SQL command:
  
  ```
  delete from ICMSTITEMEVENTS where eventcode <=600 and Created < 2002-05-01-12.00.00.000000
  ```

To reclaim the file system space after you delete the events, run the database reorganization utility on the Library Server database.

### 18.6 Removing log files

It is important to remember to remove log files on a regular basis, as long as they are no longer needed for troubleshooting or audit purposes. This prevents the log files from become overly large, taking up unnecessarily large areas of disk, and becoming unwieldy due to their sheer size. When the log files are removed, they will be recreated by the particular application that created them.

Some log files cannot be deleted while the system is in use because they are being written to. You need to stop the component that is writing to the log file in order to delete it. For a list of log files that you need to regularly check the size of, and if necessary, delete, see Appendix F, “Configuration and log files” on page 683.
It is especially important to remember to check on log files, and remove them when any form of tracing is enabled, as the log files will grow in size much more quickly than usual. See Chapter 21, “Troubleshooting” on page 559 for more information on enabling tracing.

18.7 Managing Resource Manager utilities and services

This section describes a number of utilities and processes that are installed on the Content Manager Resource Manager. The utilities are available on AIX, Linux, Solaris, and Windows. Some of the utilities exist as services on Windows. For all of the other utilities, you must log on to the server where the Resource Manager is installed. You must log on with a user ID that has DB2 administrative (DBADM) authority.

The utilities and processes include:

- The stand-alone application services: RMMigrator, RMPurger, RMReplicator, and RMStager.
- The Asynchronous Recovery utilities.
- Resource Manager/Library Server validation utility and the Resource Manager volume validation utility. These two utilities are installed with the Content Manager Resource Manager.

18.7.1 Configuration of Resource Manager utilities and services

This section provides general background information about configuring Resource Manager utilities and services.

In Content Manager, there is a central environment setup file, setprocenv.sh for UNIX or setprocenv.bat for Windows. This file stores a set of parameters for each deployed Resource Manager. These parameters are configured automatically when the Resource Manager is deployed and are used by the Resource Manager services and utilities.

Log configuration settings are specified using the logging and tracing utility in the System Administration Client.

*Environment setup file on UNIX*

UNIX includes AIX, Linux, and Solaris.

The following services and utilities depend on one central file, IBMCMROOT/config/setprocenv.sh, for environment setup:
The stand-alone application services: RMMigrator, RMPurger, RMReplicator, and RMStager.

The asynchronous recovery utilities: icmrmdel and icmrmtx

The validation utilities: icmrmlsval and icmrmvolval

The setprocenv.sh file contains one set of environment variables for each Resource Manager.

**Environment setup file on Windows**

The following utilities depend on one central file, IBMCMROOT\config\setprocenv.bat, for environment setup:

- The asynchronous recovery utilities: icmrmdel and icmrmtx
- The validation utilities: icmrmlsval and icmrmvolval
- The stand-alone application services: RMMigrator, RMPurger, RMReplicator, and RMStager, when started from the command line.

**Important:** These services are usually started as Windows services. If network-attached storage is in use, however, they must be started from the command line, and the environment setup file must be configured.

The setprocenv.bat file contains one set of environment variables for each Resource Manager.

**Environment setup file variables**

The following variables are used in the central environment setup file, setprocenv. There is one set of variables for each Resource Manager. Each variable is prefixed with the Resource Manager's identifier.

- **IBMCMROOT** DB2 Content Manager installation directory.
- **dbname** Resource Manager database name.
- **dbtype** Resource Manager database type: DB2 or Oracle.
- **rmappname** Resource Manager application name.
- **nodename** WebSphere Business Integration Server Foundation or WebSphere Application Server nodename.
- **was_home** WebSphere Business Integration Server Foundation or WebSphere Application Server home installation directory.
- **db2home** DB2 instance home directory where the Resource Manager database resides, if the Resource Manager database is a DB2 database. On Windows, enter the directory as a fully qualified path with drive letter. For example: C:\Program Files\IBM\SQLLIB. On UNIX,
enter the directory as a fully qualified path. For example: /home/db2inst1/sqlib. Leave blank if the Resource Manager database is an Oracle database.

db2_jdbc_abspath If the Resource Manager is using DB2 Type 4 connector, set this to the fully qualified path for the JDBC driver location. On Windows, enter the directory as a fully qualified path with drive letter. For example: C:\Program Files\IBM\SQLLIB\java\db2jcc.jar. On UNIX, enter the directory as a fully qualified path. For example: /home/db2inst1/sqlib/db2jcc.jar. Leave blank if DB2 Type 4 connector is not in use.

db2_jdbc_license_abspath If the Resource Manager is using DB2 Type 4 connector, set this to the fully qualified path for the JDBC license file. On Windows, enter the directory as a fully qualified path with drive letter. For example: C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cisuz.jar. On UNIX, enter the directory as a fully qualified path. For example: /home/db2inst1/sqlib/db2jcc_license_cisuz.jar. Leave blank if DB2 Type 4 connector is not in use.

orahome Oracle home installation directory, if the Resource Manager database is an Oracle database. Leave blank if the Resource Manager database is a DB2 database.

ora_jdbc_abspath Fully qualified path for the Oracle JDBC driver location, if the Resource Manager database is an Oracle database. Oracle JDBC 9.0.x is required. Leave blank if the Resource Manager database is a DB2 database.

waittime Time that the application process main thread waits for the child threads to shutdown before terminating itself.
sleeptime Time that the client must wait for the process main thread to return an acknowledgement before polling its status again.

CMRM_LOG_DIR Directory where the log file for the stand-alone application services (RMMigrator, RMPurger, RMStager and RMReplicator) is located.

CMRM_LOG_FILE Name of the log file for the stand-alone application services (RMMigrator, RMPurger, RMStager and RMReplicator).

initjavaheap Initial heap size for the stand-alone application services.

maxjavaheap Maximum heap size for the stand-alone application services.
**Resource Manager common utility parameters**

In Content Manager, the central environment setup file, setprocenv, contains information about each Resource Manager. This information is used by the Resource Manager services and utilities. You can, however, override information in the environment setup file by using the following parameters when starting services and utilities from the command line.

**Tip**: If any of the values specified in the tags contain blanks, use quotation marks to surround the values. For example, `-orajdbc "xxx yyy zzz"`.

- **-db dbname**  
  Resource Manager database (RMDB) name.  
  The parameter `-dbname dbname` is also valid.

- **-app rmappname**  
  Resource Manager application name under WebSphere Business Integration Server Foundation or WebSphere Application Server name. The parameter `-rmappname rmappname` is also valid.

- **-dbtype databasetype**  
  Resource Manager database type, where databasetype is either db2 or Oracle.

- **-was was_home**  
  WebSphere Business Integration Server Foundation or WebSphere Application Server home installation directory. The parameter `-was_home was_home` is also valid.

- **-node nodename**  
  WebSphere Business Integration Server Foundation or WebSphere Application Server nodename. The parameter `-nodename nodename` is also valid.

- **-db2home db2home**  
  DB2 instance home directory where the Resource Manager database resides, if the Resource Manager database is a DB2 database. On Windows, enter the directory as a fully qualified path with drive letter. For example: “C:\Program Files\IBM\SQLLIB”. On UNIX, enter the directory as a fully qualified path. For example: /home/db2inst1/sqlib. The parameter `-insthome insthome` is also valid.

- **-db2 jdbc db2 jdbc_abspath**  
  If the Resource Manager is using DB2 Type 4 connector, set this to the fully qualified path for the JDBC driver location. On Windows, enter the directory as a fully qualified path with drive letter. For example: “C:\Program Files\IBM\SQLLIB\java\db2jcc.jar”. On UNIX,
enter the directory as a fully qualified path. For example: /home/db2inst1/sqlibi/db2jcc.jar.

-db2_jdbc_license db2_jdbc_license_abspath If the Resource Manager is using DB2 Type 4 connector, set this to the fully qualified path for the JDBC license file. On Windows, enter the directory as a fully qualified path with drive letter. For example: “C:\Program Files\IBM\SQLLIB\java\db2jcc_license_cisuz.jar”. On UNIX, enter the directory as a fully qualified path. For example: /home/db2inst1/sqlibi/db2jcc_license_cisuz.jar.

-orahome ORACLE_HOME Oracle home installation directory, if the Resource Manager database is an Oracle database.

-orajdbc ora_jdbc_abspath Fully qualified path for the Oracle JDBC location, if the Resource Manager database is an Oracle database. Oracle JDBC 9.0.x is required.

18.7.2 Configuring the Resource Manager services on UNIX

There are four services: RMMigrator, RMPurger, RMReplicator, and RMStager. In general, the Resource Manager processes are configured using the setprocenv.sh file described in 18.7.1, “Configuration of Resource Manager utilities and services” on page 485. However, the values for dbname and rmappname can be changed if passed into the Process starting routine. These parameters override the ones that are set by the file, $IBMCMROOT/config/setprocenv.sh.

Attention: On AIX, all of the parameters, dbname, rmappname, and application, are case-sensitive. All of the process service names are registered in the /etc/services file.

Here is an example of how an entry for the services file appears:

RMMigrator_RMDB 7500/tcp #Resource Manager Migrator

In the example, RMMigrator is the stand-alone application process and RMDB is the database name. The dbname and application parameters passed to the /etc/rc.cmrmproc script should match the case in the service name registration in the /etc/services file.
18.7.3 Starting and stopping resource services on UNIX

You can start or stop a stand-alone application process as follows:

- To start all four applications:
  
  ```
  /etc/rc.cmrmproc -act start -db <dbname> -app <rmappname>
  
  Where:
  
  - `<dbname>` is the database name on which these processes are running.
  - `<rmappname>` is the name of the Resource Manager Web application.
  ```

- To stop all four applications:
  
  ```
  /etc/rc.cmrmproc -act stop -db <dbname> -app <rmappname>
  ```

- To start selective application:
  
  ```
  /etc/rc.cmrmproc -act start -db <dbname> -app <rmappname> -proc <application>
  
  Where:
  
  - `<application>` is the Resource Manager stand-alone process you want to start.
  ```

For example, to start Resource Manager migrator, RMMigrator, on database rmdb with icmrm as the Resource Manager Web application, use:

  ```
  /etc/rc.cmrmproc -act start -db rmdb -app icmrm -proc RMMigrator
  ```

- To stop selective application:
  
  ```
  /etc/rc.cmrmproc -act stop -db <dbname> -app <rmappname> -proc <application>
  ```

- To start all four applications using the default values for dbname and rmappname, specified in the $IBMCMROOT/config/setprocenv.sh file:
  
  ```
  /etc/rc.cmrmproc start
  ```

18.7.4 Asynchronous Recovery utility

Content Manager includes an automatic scheduled process called the asynchronous recovery utility. The asynchronous recovery runs at the start of each migrator cycle while the migrator is running and enabled and in its runtime window. The migrator should normally be started and enabled. It should only be excluded from running during peak load times.

The purpose is to periodically restore data consistency between a Library Server and its Resource Managers. This process is necessary for the following reasons:
To provide a rollback function for failed transactions
To complete scheduled deletion of items that are designated for deletion

The Library Server and Resource Manager can become inconsistent if the Resource Manager is down, or if communication between Information Integrator for Content and Resource Manager fails. The inconsistent state can be reconciled with the asynchronous transaction reconciliation utility.

**Attention:** Before performing any work, the migrator process will first run the Asynchronous Recovery utilities.

Another important result of running this utility is to clean up known successful transactions. As each create and update resource item transaction completes, a record is stored in the Library Server database. These records become more numerous and their database table increases in size over time. The table is cleaned up by the transaction reconciliation utility. It is important to run the utility on all of the Content Manager Version 8.1 or later Resource Managers.

Also, deleting Resource Manager resources is an asynchronous activity within Content Manager. When a user uses an application to delete an item, it is deleted, internally, from the Library Server. The asynchronous recovery deletion reconciliation utility is used to mark or physically delete the resource on the Resource Manager. Resource deletion is a multiple step process. The Resource Manager migrator, running in the background, is responsible for taking all of the resources marked for deletion and physically deleting them. Resource deletion consists of three steps:

1. An Information Integrator for Content or Content Manager application deletes an item from the Library Server.
2. The Asynchronous Recovery Deletion Reconciliation utility marks the resource for deletion on the Resource Manager.
3. The Resource Manager migrator physically deletes the resource.

Although these processes are scheduled and automatic processes, you might want to run the programs themselves, for example, as part of a database backup procedure. To do so, you need to run two separate utility programs:

- The deletion reconciliation utility (**ICMRMDEL**)
- The transaction reconciliation utility (**ICMRMTX**)

**Tip:** In a production environment, synchronize the servers prior to any system backup. This not only ensures that your databases are in a consistent state, but also removes any database entries which represent deleted documents.
Configuring the Asynchronous Recovery utility

The asynchronous recovery standalone utilities, icmrmdel and icmrmtx, take the common utility parameters and use the default values specified in the environment setup file.

Asynchronous utility logging

By default, the asynchronous utilities log to the console. You can modify the level of information logged and the location of the output in the icmrm_asyncr_logging.xml file. This XML file can be updated to output to FILE if desired. Make sure that the user ID that you use to run the utility has read permission to the XML file, and write permission to whatever log file that you configure for use.

The icmrm_asyncr_logging.xml file is installed with the Resource Manager code in the WebSphere Application Server installedApps path.

On UNIX, the default path to the file is:

```
/usr/WebSphere/AppServer/installedApps/<nodename>/icmrm.ear
/icmrm.war/icmrm_asyncr_logging.xml
```

On Windows, the default path is:

```
x: WebSphere\AppServer\installedApps\<nodename>\icmrm.ear\icmrm.war
\icmrm_asyncr_logging.xml
```

<nodename>: WebSphere Business Integration Server Foundation or WebSphere Application Server nodename.

Running the Asynchronous Recovery utilities on Windows

You run the asynchronous recovery utilities from a command prompt using two of the common utility parameters.

- To run the deletion reconciliation utility:
  a. Change to the IBMCMROOT\bin directory.
  b. Enter:
     `icmrmdel.bat -db dbname -app rmappname`

- To run the transaction reconciliation utility:
  a. Change to the IBMCMROOT\bin directory.
  b. Enter:
     `icmrmtx.bat -db dbname -app rmappname`
Running the Asynchronous Recovery utilities on UNIX

The asynchronous recovery utilities run when you start the migrator. You can also run the asynchronous recovery utilities from a command prompt using two of the common utility parameters. You must be logged in as the root user to run them manually.

- To run the deletion reconciliation utility:
  - Change to the IBMCMROOT/bin directory.
  - Enter:
    
    ```
    ./icmrmdel.sh -db dbname -app rmappname
    ```

- To run the transaction reconciliation utility:
  - Change to the IBMCMROOT/bin directory.
  - Enter:
    
    ```
    ./icmrmtx.sh -db dbname -app rmappname
    ```

Tip: After running the Asynchronous Recovery utilities, run the RUNSTATS function on your databases to ensure that they are operating efficiently. See 18.2, “Optimizing server databases” on page 472 for help in using this command.

18.7.5 Validation utilities overview

The purpose of the validation utilities is to analyze discrepancies between three components: the Library Server, the Resource Manager, and the storage system(s) used by the Resource Manager through its defined device managers. Any of these components can fail and require a restoration via a backup that may be out of synchronization with the other two components.

Because there is no direct link between the Library Server and the storage system, (an example of a storage system is VideoCharger or Tivoli Storage Manager), differences must be reported between the Library Server and the Resource Manager, and the Resource Manager and the storage system using the following utilities:

- The Resource Manager/Library Server validation utility (`icmrmlsval.sh` or `icmrmlsval.bat`) generates reports that describe discrepancies between the Library Server and the Resource Manager.

- The Resource Manager volume validation utility (`icmrmvolval.sh` or `icmrmvolval.bat`) generates reports on discrepancies between the Resource Manager and the storage system.
The reports are in XML. You can use commonly available XML tool or browser to view or manipulate the utility output files. Content Manager installs the XML document type definition (DTD) required by the validation utility output files.

You can modify the two utility files with information specific to your Content Manager system. The validation utilities are located in the bin directory in the Resource Manager installation directory.

The validation utility creates and drops a temporary DB2 table. The environment script requires the resource database name, user ID, password, schema, Web application path, and DB2 instance. To set the environment for both validation utilities, type:

setenvproc.bat or setenvproc.sh.

Logging
By default, the validation utilities log to a file named icmrms.validator.log file in the WebSphere logs directory. You can modify the level of information logged and the location of the output in the icmrms_validator_logging.xml file. Be sure that the user ID that you use to run the utility has read permission to the XML file, and write permission to whatever log file that you configure for use.

The icmrms_validator_logging.xml file is installed with the Resource Manager code in the WebSphere Application Server installedApps path. On AIX, the default path to the file is:

/usr/WebSphere/AppServer/installedApps/<nodename>/icmrms.ear/icmrms.war/icmrms_validator_logging.xml

On Solaris, the default path is:

/opt/WebSphere/AppServer/installedApps/<nodename>/icmrms.ear/icmrms.war/icmrms_validator_logging.xml

On Windows, the default path is:

x:\WebSphere\AppServer\installedApps\<nodename>\icmrms.ear\icmrms.war\icmrms_validator_logging.xml

<nodename>: WebSphere Business Integration Server Foundation or WebSphere Application Server nodename

18.7.6 Resource Manager/Library Server validation utility
The Resource Manager/Library Server validation utility queries the Library Server for all of the objects created or updated in a specified time period. It then searches the Resource Manager database and detects any discrepancies. The utility runs on the Resource Manager server and requires connectivity to the Library Server database.
To start the utility, navigate to the Resource Manager bin directory and type:

```
icmrmlsval.sh or icmrmlsval.bat
```

The utility requires input parameters that are described in Figure 18-2. Both dashes (-) and forward slashes (/) are handled as the parameter separator. The parameter tags are supported in both lower and upper case.

**Table 18-2 Resource Manager/Library Server validation utility parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-B YYYY-MM-DD-HH.MM.SS</code></td>
<td>The beginning time and date of the objects to examine. Use this parameter with the <code>-E</code> parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects prior to the <code>-E</code> date are returned, or all of the objects are returned if <code>-E</code> is also not defined.</td>
</tr>
<tr>
<td><code>-E YYYY-MM-DD-HH.MM.SS</code></td>
<td>The ending time and date of the objects to synchronize. Use this parameter with the <code>-B</code> parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects after the <code>-B</code> date are returned, or all of the objects are returned if <code>-B</code> is also not defined.</td>
</tr>
<tr>
<td><code>-F output-path</code></td>
<td>The absolute path to be used for the output files. The utility creates the UTF-8 XML files in this directory. This parameter is required.</td>
</tr>
<tr>
<td><code>-H</code></td>
<td>This parameter displays help information about how to invoke the utility. All other parameters are ignored and no processing occurs.</td>
</tr>
</tbody>
</table>

The utility creates a temporary table, RMLSITEMS used to accumulate object statistics for the validation. At the end of the validation, this table is normally dropped. If the utility determines that the table is present, it presumes another version of the utility is operating, and exits. If the table is left behind due to an aborted run, you need to drop this table. Connect to the Resource Manager database and drop the table with the following command:

```
db2 drop table RMLSITEMS
```

The following line shows an example of how to invoke the Resource Manager/Library Server utility on an AIX server:

```
./icmrmlsval.sh -F /reportsdirectory -B 2002-08-30-00.00.00 -E 2002-09-01-00.00.00
```

Chapter 18. Maintenance 495
Understanding the Resource Manager/Library Server reports

The base file names of the reports are “icmrmlsval YYMMDDHHMMSS _”+Report Type string +”.xml”. The Report Type string identifies the type of discrepancies a report contains. The description of the different report types are detailed in this section. The timestamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- cmrmlsval20020531123456_ORPHAN.xml
- cmrmlsval20020531123456_NOTINRM.xml
- cmrmlsval20020531123456_SIZEMISMATCH.xml
- cmrmlsval20020531123456_COLLECTIONMISMATCH.xml
- cmrmlsval20020531123456_DATEMISMATCH.xml

There are several types of Resource Manager/Library Server reports:

Orphan Entries are added to the ORPHAN report if an object is on the Resource Manager, but the Library Server does not have a reference to the object. The report contains information about the object from the Resource Manager database.

Not in RM Entries are added to the NOTINRM report if the Library Server has a reference to an object, but the object is not on the Resource Manager. The report contains information about the object from the Library Server database.

Size mismatch Entries are added to the SIZEMISMATCH report if the size of an object on the Library Server does not match the size of an object on the Resource Manager. The report contains information about the object from the Resource Manager and Library Server databases.

Collection mismatch Entries are added to the COLLECTION report if the collection of an object on the Library Server does not match the collection of an object on the Resource Manager. The report contains information about the object from the Resource Manager and Library Server databases.
Date mismatch Entries are added to the DATEMISMATCH report if the object update date on the Library Server does not match the object update date on the Resource Manager. Under normal circumstances, if there is any synchronization problem between the Library Server and the Resource Manager, the object update date does not match. In order to reduce redundant entries in the different reports, entries are not added to the DATEMISMATCH report if they have been added to the collection mismatch or size mismatch reports. The report contains information about the object from the Resource Manager and Library Server databases.

18.7.7 Resource Manager volume validation utility

The Resource Manager volume validation utility checks each object in its database that was added or changed in a specified date range. It queries the device manager for the attributes of that object and generates reports for each object whose information in the database is different than reported by the device manager. You may want to use the utility if you have a restore data on a volume after a volume crash. The utility helps you to verify that the data is restored correctly. The Resource Manager must be running when you use the utility.

Tip: Use the Resource Manager volume validation utility during times of low traffic on the Resource Manager.

The validation utility does not search the storage system for orphaned objects (objects not referenced by the Resource Manager). Because there are a wide variety of storage systems that are often used for storing files other than those managed by Content Manager, the scanning for orphaned files can be extremely time consuming and may produce a large quantity of false positives.

The Resource Manager volume validation utility runs on the Resource Manager server and only requires access to its own database and the device managers responsible for the volumes that are being checked.

Starting the Resource Manager volume validation utility

The Resource Manager volume validation utility is icmrmvolval.sh or icmrmvolval.bat. To start the utility, navigate to the bin directory in the Resource Manager home directory.

The Resource Manager volume validation program uses specific input parameters (see Table 18-3 on page 498). Both dashes (-) and forward slashes (/) are handled as the parameter separator. The parameter tags are supported in both lower and upper case.
### Understanding the validation discrepancy reports

The base file names of the discrepancy reports are 
“icrmrvolvalYYMMDDHHMMSS_" + Report Type string +".xml”. The Report Type string identifies the type of discrepancies a report contains. The description of the different report types are detailed later in this section. The timestamp allows the administrator to run the utility multiple times without overwriting the output files. Examples of default names with the default report type are:

- cmrmrval20020531123456_FILENOTFOUND.xml
- cmrmrval20020531123456_SIZEMISMATCH.xml

There are two default discrepancy reports:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-B YYYY-MM-DD-HH.MM.SS</td>
<td>The beginning time and date of the objects to examine. Use this parameter with the -E parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects prior to the -E date are returned, or all of the objects are returned if -E is also not defined.</td>
</tr>
<tr>
<td>-E YYYY-MM-DD-HH.MM.SS</td>
<td>The ending time and date of the objects to synchronize. Use this parameter with the -B parameter to restrict the number of objects that the utility must examine. This parameter is optional. If it is not present, all of the objects after the -B date are returned, or all of the objects are returned if -B is also not defined.</td>
</tr>
<tr>
<td>-F output-path</td>
<td>The absolute path to be used for the output files. The utility creates the UTF-8 XML files in this directory. This parameter is required. If the files currently exist, they are overwritten.</td>
</tr>
<tr>
<td>-H</td>
<td>This parameter causes the program to display help information about how to invoke the utility. All of the other parameters are ignored and no processing occurs.</td>
</tr>
<tr>
<td>-V volume-name</td>
<td>The logical volume name on which you want to perform the validation. Use this parameter to limit the number of storage systems to one volume. This parameter is optional. If not used, all storage systems are searched.</td>
</tr>
</tbody>
</table>
Entries are added to the FILENOTFOUND report if an object is in the Resource Manager database, but it is not found on the volume recorded in the database. A file is considered “not found” if the volumes device manager either reports that the file does not exist, or reports that the file has a zero file size when the size in the database is non-zero. The report contains the object information from the Resource Manager database.

Entries are added to the SIZEMISMATCH report if the size of an object in the Resource Manager database does not match the size reported by the device manager. The report contains the object information from the Resource Manager database and the size reported by the device manager.

**18.8 Replacing or repartitioning a hard disk**

If a volume or file system that is used by your Resource Manager becomes full, you can replace or repartition the physical disk on which it is located to make more space available.

Replacing or repartitioning the disk makes the information stored in the volumes table (RMVOLUMES) for that volume or file system invalid. When updating the Resource Manager volumes, do not run the destager at any point of this process. Otherwise, the volumes will not be the same. Use the following procedures to update the information in the volumes table.

**18.8.1 Replacing the staging volume for UNIX**

The directory for the staging volume is in the Resource Manager database table, RMSTAGING. Follow these steps to replace the staging volume:

1. Change the permissions on the new staging directory to match those of your Resource Manager ID or what is currently in place for the existing staging directory

2. If all files in existing staging directory are currently read-writeable, you can skip this step because these files have been destaged already; otherwise, copy all existing files to the new staging volume:

   \texttt{cp -rp current\_staging\_directory new\_staging\_directory}

3. Update the location of your staging volume in the Resource Manager database. Open a DB2 command prompt and enter the following commands, each on a new line:
db2 "connect to <RM db> user <user ID> using <password>"
  db2 "update rmstaging set sta_path=staging_path"

Where:

- <RM db> is the Resource Manager database (in our scenario, it is rmdb).
- <user ID> is the user ID (in our scenario, icmadmin) used to connect to the Resource Manager database.
- <password> is the password for the user ID.
- <staging path> is the location of the staging directory, as an absolute path with the trailing slash.

18.8.2 Replacing the storage volume for UNIX

The Resource Manager uses the `vol_path + the string_table value of lbosdata + collection + num_bucket_value` to develop the path. The logical_volume and mount_point are used in various calls to get file system information. Follow these steps to update the Resource Manager storage volume:

1. Stop the Resource Manager.
2. Change the permissions on the new staging directory to match those of your Resource Manager ID or what is currently in place for the existing staging directory.
3. Copy all existing files to the new storage volume:
   ```
   cp -rp current_storage_directory new_storage_directory
   ```
4. Update the location of your storage volume in the Resource Manager database. Use `df -k` to determine the FILESYSTEM and MOUNTED ON location for new storage directory. To update the storage volume, enter the following commands, each on a new line:
   ```
   db2 "connect to <RM db> user <user ID> using <password>"
   db2 "select vol_volumeid,vol_logicalname,vol_mountpoint from rmvolumes"
   ```

Where:

- <RM db> is the Resource Manager database (in our scenario, it is rmdb).
- <user ID> is the user ID (in our scenario, icmadmin) used to connect to the Resource Manager database.
- <password> is the password for the user ID.
5. Determine which VOLUMEID is the one you need to change. For example, to change VOLUMEID=1, the new logical volume is /dev/data1, and mount point is /rm/data1, enter:

   db2 "update rmvolumes set vol_logicalname='/dev/data1 'where vol_volumeid=1"
   db2 "update rmvolumes set vol_mountpoint='/rm/data1 'where vol_volumeid=1"
   db2 "update rmvolumes set vol_size=0 where vol_volumeid=1"
   db2 "update rmvolumes set vol_path='/rm/data1 'where vol_volumeid=1"
   db2 "update rmvolumes set vol_freespace=0 where vol_volumeid=1"


   Notice that the latter two steps force the Resource Manager to recalculate the volume space and capacity during any new stores. These values are reflected in the RMVOLUMES tables when the Resource Manager shuts down.

### 18.8.3 Replacing the staging volume for Windows

The directory for the staging volume is in the Resource Manager database table (RMSTAGING). Follow these steps to replace the staging volumes:

1. Change the permissions on the new staging directory to match those of your Resource Manager ID or what is currently in place for the existing staging directory.

2. If all files in existing staging directory are currently read-writeable, you can skip this step because these files have been destaged already; otherwise, copy all existing files to the new staging volume:

   `copy -rp current_staging_directory new_staging_directory`

3. Update the location of your staging volume in the Resource Manager database. Open a DB2 command prompt and enter the following commands, each on a new line:

   ```
   db2 "connect to <RM db> user <user ID> using <password>"
   db2 "update rmstaging set sta_path=<staging path>"
   ```

   Where:

   - `<RM db>` is the Resource Manager database (in our scenario, it is rmdb).
   - `<user ID>` is the user ID (in our scenario, icmadmin) used to connect to the Resource Manager database.
• `<password>` is the password for the user ID.
• `<staging path>` is the location of the staging directory, as an absolute path with the trailing slash.

### 18.8.4 Replacing the storage volume for Windows

If you replace or repartition the hard disk that contains the LBOSDATA directory, you need to identify the new configuration to your system:

1. Stop the Resource Manager.
2. Restore the LBOSDATA directory to the new disk or partition.
3. Open a DB2 command prompt.
4. Manually edit the volumes table to change the following columns to zero for the volume that has been changed. Enter each command on a new line:

   ```sql
   update rmvolumes set vol_size=0 where vol_volumeid=ID
   update rmvolumes set vol_freespace=0 where vol_volumeid=ID
   ```

   Where:
   - `<ID>` is the volume ID.

5. The next time the Resource Manager writes or deletes an object, the information is read from the new disk or partition and placed in the volumes table.

6. If your volume is on a different partition, then manually edit the RMVOLUMES table to update the VOL_LOGICALNAME and VOL_MOUNTPOINT.

   For example, assume the volume you wish to replace is defined in the RMVOLUMES table entry with VOL_VOLUMEID=1. If your new partition is F and this partition is labeled FDRIVE, enter:

   ```sql
   UPDATE RMVOLUMES set VOL_LOGICALNAME='FDRIVE' where vol_volumeid=1
   UPDATE RMVOLUMES set VOL_MOUNTPOINT='f:' where vol_volumeid=1
   ```

7. Start the Resource Manager.

### 18.9 Backup

This section covers backup of the Content Manager system for both the Multiplatforms and z/OS platform.
18.9.1 Backup for Content Manager Multiplatforms

For Content Manager on Multiplatforms, it is important to back up four key components:

- The Library Server database: Use your database manager tools to facilitate this.
- The Resource Manager database: Use your database manager tools to facilitate this.
- The LBOSDATA directory on every Resource Manager.
- Tivoli Storage Manager (TSM) volumes. It is important to remember to back up any data that is migrated to TSM via Content Manager migration policies; otherwise, you will have a single point of failure and data loss within your system. This can be accomplished by using TSM copy storage pools, which may be made up of tape volumes which can be stored off site.

It is not necessary to back up objects within the staging area, as all of the objects within this directory exist else where in the system, and hence already have been backed up if you use the above guideline.

With these four key components, you can rebuild your Content Manager system, even if the original server is completely destroyed.

If you choose to back up only the four components listed above (as opposed to a full system backup), you need to reinstall the Content Manager code, and its software prerequisites onto another machine, should the original machine be destroyed, in order to restore your system. Not only should this time taken to reinstall a system be taken into consideration when forming a recovery plan, you must also make sure you have easy access to the installation media, via the original CDs or via a network drive.

If at all possible, perform full backups for each of your Content Manager servers. A hierarchical storage management product such as TSM is ideal for this. When choosing the type of media to back your system up to, consider the relative speed of the media. For example, restoring a DB2 database backup that spans multiple magnetic tapes takes much more time than restoring the same database backup from a fiber attached storage area network (SAN).

TSM can also be used to back up database backup images and database logs. These backups can be stored on any type of media that TSM supports; therefore, it is possible to back up DB2 archive logs to tape volumes on TSM automatically, reducing the amount of storage space needed on the server running DB2. DB2 can be integrated with TSM so that DB2 commands can be executed as follows:

```
db2 backup database icmnlisdb use tsm
```
For information on how to integrate DB2 and TSM to provide this type of functionality, refer to the IBM Redbook *Backing Up DB2 Using Tivoli Storage Manager*.

Make sure that you back up all components of the Content Manager system together. If you need to restore the system later, each component must be from the same point in time.

1. Identify the LBOSDATA areas. Execute the appropriate query for your operating system:
   UNIX `select vol_mountpoint from rmvolumes`
   Windows `select vol_logicalname from rmvolumes`

2. Pause the system.

3. Perform the backups. Back up:
   - Library Server database
   - Resource Manager database
   - LBOSDATA areas
   - Data stored in Tivoli Storage Manager

   If possible, before backup database, perform `db2stop/db2start` to ensure there are no clients or services connecting to the database in order to perform full backup. Or, for DB2 UDB V8.2 or higher, use `quiesce` command, for more detailed information about this command, refer to *IBM DB2 Universal Database Command Reference*, SC09-4828.

4. Resume the system.

**Pausing DB2 Content Manager for backups**

The Library Server PAUSESERVER utility enables you to stop all Content Manager transaction processing in preparation for Library Server and Resource Manager backup processes.

To pause Content Manager, run PAUSESERVER, specifying a future time (UTC). When the system time is equal to or greater than the time that you specify, the Library Server will block all new transactions.

If there are transactions processing when the pause time is reached, those transactions will run until completion if they do not exceed the value of MAXTXDURATION. MAXTXDURATION is a column of ICMSTSYSCONTROL table. It is a numeric, which points to the maximum duration by second. If a transaction that is processing exceeds the maximum time allowed, it is cancelled and all work owned by the transaction is rolled back.

When all transactions have completed on the Library Server, there will be no client-initiated actions to any Resource Manager, thereby suspending Content Manager and leaving you free to create a consistent backup of all Content Manager servers.
To pause the Library Server, follow these steps:

1. Open a DB2 command prompt.
2. Change to the IBMCMROOT\bin directory.
3. Enter the version of the command for your operating system:
   - UNIX:
     ```bash
     ./pauseserver.sh dbname userid password SUSPENDSERVERTIME
     ```
   - Windows:
     ```bat
     pauseserver.bat dbname userid password SUSPENDSERVERTIME
     ```

   This command updates the SUSPENDSERVERTIME field in the ICMSTSYSCONTROL table. When that time is less than or equal to the current time, all new transactions are rejected. If an application is storing an object to a Resource Manager, those operations will complete if they can do so within the time specified in MAXTXDURATION in ICMSTSYSCONTROL. After that time, all requests to the Library Server are rejected.

   For example, if want to pause Content Manager server on windows platform at 2005-12-14-16:42:00:000000, the local time. The Library Server database is icmnlsdb; userid is icmadmin; password is password, then run below command:

   ```bat
   pauseserver icmnlsdb icmadmin password 2005-12-14-16:42:00:000000
   ```

   **Resuming DB2 Content Manager after backups**

   The RESUMESERVER utility enables you to resume transaction processing. To resume Content Manager, run RESUMESERVER, which will update SUSPENDSERVERTIME to null and resume transaction processing.

   To resume Library Server processing, follow these steps:

   1. Open a DB2 command prompt.
   2. Change to the IBMCMROOT\bin directory.
   3. Enter the version of the command for your operating system:
      - UNIX:
        ```bash
        ./resumeserver.sh dbname userid password
        ```
      - Windows:
        ```bat
        resumeserver.bat dbname userid password
        ```
18.9.2 Backup of z/OS DB2 databases

As all of the data is stored in DB2, you have to implement a proper backup strategy for you Content Manager database. Usually a backup strategy is already defined at z/OS customers and backups are often performed as incremental backups on a daily basis. For Content Manager, there is nothing special to consider as long as your system does not deal with millions of documents per day.

For a system that is dealing with a large amount of data, special planning for the backup strategy is required. Ask your IBM service representative for assistance.

18.9.3 Backup of OAM DB2 tables

For OAM, it may be necessary to set up a different backup strategy, depending on your storage strategy. If you are planning to keep the objects on DASD for a long time, you have to handle very large tables. As mentioned earlier, special configuration of your storage strategy may be useful to manage these tables.

For example, you can define a storage group, which is connected to one object database, for a specific time frame, such as one month. After this period, you create a new storage group, connected to another object database. If the OAM table does not change, then frequent (such as weekly) DB2 full image copies is not required. This reduces the amount of data and the time needed for the backup process.

This is just one idea. There are many more ways to get it done, depending on your requirements and your installation.
18.10 Maintenance review

Even though this chapter has gone into detail on some maintenance related subjects, the key points to remember are as follows:

- Monitor object storage space. This includes local and remote storage devices.
- Make sure the Resource Manager and Library Server databases are optimized regularly.
- Make sure the Resource Manager utilities and services that you need in your environment are running and performing their job as expected.
- Regularly delete log files that you do not need anymore, to prevent them taking up space unnecessarily. This is particularly important when you enable tracing, as the log files will grow in size extremely quickly.
- Back up critical components on a regular basis, and test that these backups work on a regular basis by restoring your system to another machine.

Special note for z/OS maintenance tasks

Maintenance tasks for Content Manager z/OS are almost the same as described for the Multiplatform product.

The main difference in maintenance is that, in Content Manager z/OS, the Resource Manager uses OAM to store the objects. There is no native Content Manager data storage. OAM also manages the migration of objects during the entire life cycle. Only the collection is defined in the Content Manager. This collection must reflect to a collection in your OAM system; otherwise, the object cannot be stored in OAM.

Most tasks regarding Resource Manager usually do not apply to a z/OS Resource Manager, except, for example, in the case of Asynchronous Recovery utility.
Export and import utilities

This chapter covers the process of exporting and importing data from XML readable files and the main APIs used for exporting and importing Content Manager system administrative objects and metadata.

With this new export and import feature, you can easily integrate or move objects from one Content Manager system to another on the same platform or across different platforms.
19.1 XML export and import in system administration

A new feature introduced with Content Manager V8.3 is the system administration enhancement permitting XML-based export and import.

With this feature, you can use System Administration Client and select one or more objects from one Content Manager system, export them to a file, and then import the data to another Content Manager system, or the same system at a later time. This is one way to back up your Content Manager system configuration.

The exported files are in readable XML format. You can edit the exported files, add, remove objects, modify objects’ names and other properties.

When importing definitions into an identical, clean system, it is straightforward. If the import data does not match with the configuration of the target system, or the objects already exist in the target system, you may encounter problems. For this reason the System Administration Client provides an interactive option on import that shows the information which will be imported beforehand, and provides some control over what is imported.

19.2 Exporting data as XML

You can select one or more objects and export them to a readable XML file or directly to another server. By exporting data as XML, you can transfer Content Manager metadata, including data model objects, such as item types and their attributes, and administration objects, such as server definitions and access control lists, from one Content Manager system to another.

You can export objects with their prerequisites. Each metadata object has a set of attributes. Some of these attributes might be other Content Manager objects. These other objects are considered prerequisite, or dependent, objects.
Important: For a single export action, you cannot select objects of different types. For example, you cannot select to export some item types and some access control lists in the same export action. You can export these objects in two separate export actions.

Certain characters used for the name of a data model object for Content Manager or WebSphere Information Integrator for Content are not valid in the XML context. For example, XML does not allow “XML” to be at the beginning of an element or attribute name. Therefore, an item type name such as “XMLDocument” cannot be directly mapped to an element name in XML. Another example is that an XML name does not allow spaces. Therefore, a federated entity named “project entity” cannot be exported directly as named in the XML element. The same rule applies to a federated attribute.

To export data:

1. In the navigation pane, select the object you want to export (Figure 19-1).
2. If you select one or more objects of the same type, right-click and select **Export to XML** (Figure 19-2).

Figure 19-2 Selecting the Export XML option

The Export Options window displays (Figure 19-3).

Figure 19-3 Export Options window
3. If you select a single tree node, right-click and select Export All to XML (Figure 19-4).

4. Under Dependent definitions to also export, choose the dependent objects you want to export. For example, if you are exporting an item type (Figure 19-2), it might have a dependency on an access control list or an attribute group. If you want the dependent objects exported, select the appropriate check box.

5. Under Export destination, you can export your metadata to a file or to another server:
   a. To a file. Export your data directly to a file:
      - Browse to the directory you want to store the file
      - Enter a file name

   **Important:** The file containing data model objects has a file extension of .xsd. A file containing administrative objects has an extension of .xml.

   You can import the file after it is created to another system using the System Administrative option menu Tools → Import XML.
b. **To another server.** Export your data directly to another server that is defined to the System Administration Client:

- Select the server name from the list.
- Select your export preference:

  If you want to compare the definitions you import to the definitions that already exist in the system, select **Process interactively**. We recommend using **Process interactively** only when the XML file being imported is a *small one*. If it is a huge XML file, it might take a long time because you might have to resolve many conflicts for each definition.

  Otherwise, choose **Log results to XML import log**. If an error occurs, it will be logged and the processing will continue with the next object.

6. Click **OK** to export the selected objects. The Export Progress window displays (Figure 19-5), showing the status of your action.

![Export Progress window](image)

*Figure 19-5  The Export Progress window*

To properly import a particular object, all of the prerequisite objects have to either exist or be already imported to the system. To guarantee that this occurs, the export order is important. When you choose to export an object with the prerequisite option on, the proper order is ensured. However, Content Manager cannot handle the situation where there is a cyclic dependency among objects of the same type.

For example, suppose that there are three item types: A, B, and C. Here is how they are related to each other:

- Item Type A depends on Item Type B (because of a foreign key definition)
- Item Type B depends on Item Type C (because of an auto foldering definition)
- Item Type C depends on Item Type A (because of a foreign key definition)

A warning message is logged in the connector log file as well as displayed in the System Administration Client when this situation is detected during XML Export.
The message in the log file describes where the cycle is. In our example, the following log message is found in the log file:

```[MSG]: There is a cycle ([A, B, C]) within the dependent objects of A of type ITEM TYPE. When import the definitions to another system, please remove the cycle before import the XML document.```

To work around this problem, you should follow these steps:

1. Make a copy of the exported XML file.
2. Break the cycle in the XML file. In our example, you would remove the foreign key definition temporarily from C to A in the XML export file.
3. Import the modified XML file.
4. Add the removed definition back to the XML file. In our example, it is the foreign key definition from C to A.
5. Import only the objects being affected. In our example, it is the item type C.

During the export process, the real password of a system administration object, such as a user, will not be exported. The default text “password” will be used instead. This feature is introduced for security reasons. There should not be any real password in clear text written in the export file.

The following system admin object passwords will be exported as “password”:

- The password for user
- The password in Resource Manager’s server definition (in CMResourceManagerDefinitions)
- Resource Manager password in Resource Manager configuration definition
- DB2 Text Information Extender or DB2 Net Search Extender password for Library Server configuration

When the export process is finished, these object will have “password” as their passwords.

If you want to import the exported object back to the database, you should change the default password in the exported file before importing it. Otherwise, the default password (“password”) will be imported into the target system. This applies to all the system administration objects that have password fields.

Because the password exported is always the default text “password”, while importing in interactive mode, the comparison is based on the target system having the default password as well. Even though the passwords in the source object and target system object are the same, the conflict might still arise.
19.3 Importing data as XML

You can import one or more data models, such as attributes, attribute groups, or item types; or system administration objects, such as users, privileges, and access control lists from a readable XML file. You can also use this feature in conjunction with the Export menu to move metadata objects from one system to another.

Because the exported files are in readable XML format, you can edit them, adding or removing objects or modifying their names or other properties. When importing definitions into an identical, clean system, it is unlikely that there will be problems. However, in cases where the import data does not match the configuration of the system, or there is existing configuration information, various problems can arise. For this reason the System Administration Client provides an interactive option on import which will be imported beforehand, and provides some control over what is imported.

**Restriction:** Certain characters used for the name of a data model object for Content Manager or WebSphere Information Integrator for Content are not valid in the XML context. For example, XML does not allow “XML” to be at the beginning of an element or attribute name. Therefore, an item type name such as “XMLDocument” cannot be directly mapped to an element name in XML. Another example is that XML name does not allow spaces. Therefore, a federated entity named “project entity” cannot exported directly as named in the XML element. The same rule applies to a federated attribute.
To import data from a readable XML file, proceed as follows:

1. From the main menu, click **Tools → Import XML**. The Import XML Options windows displays (Figure 19-6).

![Figure 19-6 Selecting the Import option](image1)

2. Click **Browse** to select the file from which you want to import the objects, either an .xsd file for data model objects or an .xml file from administrative objects (Figure 19-7).

![Figure 19-7 Browsing the files to import](image2)
3. Select your import preference (Figure 19-8):

To compare the definitions you want to import to the definitions that may already exist in the system, select **Process interactively**. Note that the select process occurs interactively when the XML file being imported is a small one. If it is a huge XML file, it might take a long time because you might have to resolve many conflicts for each definition.

Otherwise, choose **Log results to XML import log**. If an error occurs, it will be logged and the processing will continue with the next object.

![Image](image.png)

**Figure 19-8 Selecting the import preference**

4. Click **Import** to begin the import process.

If you receive one of the following error messages:

DGL0683A: Internal error: The root element required
‘CMResourceManagerDefinitions’ is not unique in the source system or file

or

DGL0683A: Internal error: The root element required
‘CMSSystemAdminDefinitions’ is not unique in the source system or file

You might be attempting to import to the incorrect server type.

For example, if you received the first error message, you have a Resource Manager selected but the import file contains Library Server definitions. To avoid these errors, select the appropriate server name, then select **Tools → Import XML**.

If you received the second error message, you might be trying to import WebSphere Information Integrator for Content objects that are not supported for XML import.
The following types of WebSphere Information Integrator for Content objects are supported for XML import:

- Server configuration
- Federated entity
- Search template

When you click **Import**, the import files are parsed, and the objects found within are compared to existing objects (if any) on the server. The results are then shown graphically in the Import Preprocessor Results window.

### 19.3.1 Understanding the Import Preprocessor Results window

In the Import Preprocessor window, a tree shows each type of object which is to be imported, and the children of these tree nodes represent the individual objects which will be imported. In the example shown in Figure 19-9, the type **Attributes** contains objects **Address**, **Customer**, **Phone**, while the type **Item types** contains just **S**. Each these tree nodes has a state, illustrated both by the icon used and the icon's tool tip. Initially only two states (**Same** and **New different**) are presented, however five additional states (**New**, **Different**, **Do not import**, **New different checked** and **Different checked**) can result from actions you take on these nodes.

![Import Preprocessor Results window](image)

*Figure 19-9  Import Preprocessor Results window (Part 2)*
Figure 19-10 shows each of the seven possible states and its meaning when applied to an object node.

<table>
<thead>
<tr>
<th>Number</th>
<th>Tool tip</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>name</em> (Do not import)</td>
<td>![Icon]</td>
<td>Object will not be imported</td>
</tr>
<tr>
<td>2</td>
<td><em>name</em> (Same)</td>
<td>![Icon]</td>
<td>Object exists and all properties are identical</td>
</tr>
<tr>
<td>3</td>
<td><em>name</em> (New)</td>
<td>![Icon]</td>
<td>Object does not yet exist and validated properties are acceptable</td>
</tr>
<tr>
<td>4</td>
<td><em>name</em> (New different checked)</td>
<td>![Icon]</td>
<td>Changes to a new different object are accepted</td>
</tr>
<tr>
<td>5</td>
<td><em>name</em> (Different checked)</td>
<td>![Icon]</td>
<td>Changes to a different object are accepted</td>
</tr>
<tr>
<td>6</td>
<td><em>name</em> (New different)</td>
<td>![Icon]</td>
<td>Object does not yet exist and one or more validated properties have been changed</td>
</tr>
<tr>
<td>7</td>
<td><em>name</em> (Different)</td>
<td>![Icon]</td>
<td>Object exists but properties are different</td>
</tr>
</tbody>
</table>

*Figure 19-10 Preprocessor tree node states*

The state assigned to a type node is the highest state any of its children, using the sequence in Figure 19-10. You will that the Continue button remains disabled as long as any of the nodes has a state greater than 5.

This is because New different and Different nodes represent objects which either are not going to be imported as you expect, or will change existing system objects. In both cases, you should inspect the changes and accept them, changing the state to New different checked or Different checked.

### 19.3.2 Additional details on each state

*Do not import* means that the object, or all objects of this type, will be ignored during import. To set this state, either choose Do not import from the pop-up menu for the node, or click Do not import while viewing details. If you later select Import, the object will return to its previous state (Figure 19-11).
Same means that the object already exists in the target system, and that all of its properties are identical to those in the import file. This does not mean that dependent objects are identical, although properties refer to them. As a typical example, consider an item type object which contains properties which refer to attributes. Same means the attribute names are identical, but does not mean the properties of the attribute object itself are all the same. This could lead to problems when the object is imported. For example, suppose that a dependent attribute already exists and is an integer type, but the item type being imported expects it to be a character type and sets it to be text searchable - naturally an error will occur. Generally it is a good idea to import such dependencies first even if they already exist, in order to be certain that they are correct.

New means that the object does not yet exists in the target system and that no problems were found with properties which are validated. However, note that only a few properties are validated, and that these are only concerned with the existence of dependence objects, not the correctness of the property’s value. See the discussion of New Different for more details.

New different checked means that an object which was in the New different state has been checked, in other words the differences have been accepted, so it is okay to import.
To do this, first view the details (Figure 19-12), then click **Accept** in the Details of Import Definition and Target Definition window (Figure 19-13).

![Figure 19-12  View Details option](image1.png)

![Figure 19-13  Details of Import definition and target definition window](image2.png)

*Different checked* means that an object which was in the *Different* state has been checked, so it is okay to import.
New different means that the object does not yet exist in the target system and that one or more problems were found with properties which are validated. You should view the details and accept the changes which were made to the import data, if appropriate. You can also modify them, if there is more than one possible value. Keep in mind that only a few properties are validated (see Table 19-1).

These properties are ones which are references to dependent objects that are very likely to be different from one system to another. Primarily they are Resource Managers, workstations collections, and ACLs. The names referred to by these properties will be checked against both the target system and the objects to be imported, and if not found, the property will be changed to a valid value for you.

Table 19-1 Validated properties

<table>
<thead>
<tr>
<th>Object type</th>
<th>Property name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library server configurations</td>
<td>Library ACL name</td>
</tr>
<tr>
<td>Users</td>
<td>Default Resource Manager</td>
</tr>
<tr>
<td>Users</td>
<td>Default SMS collection</td>
</tr>
<tr>
<td>Users</td>
<td>User ACL</td>
</tr>
<tr>
<td>Item types</td>
<td>Access control list</td>
</tr>
<tr>
<td>Item types</td>
<td>Resource manager</td>
</tr>
<tr>
<td>Item types</td>
<td>Collection</td>
</tr>
<tr>
<td>Item types</td>
<td>Prefetch collection</td>
</tr>
<tr>
<td>Work nodes</td>
<td>Access control list</td>
</tr>
<tr>
<td>Work nodes</td>
<td>Folder item type</td>
</tr>
<tr>
<td>Work nodes</td>
<td>Required item type</td>
</tr>
<tr>
<td>Processes</td>
<td>Access control list</td>
</tr>
<tr>
<td>Worklists</td>
<td></td>
</tr>
</tbody>
</table>

Different means that the object already exists in the target system, and that one or more of its properties is different from that in the import data. One of four outcomes is possible for each property of the object:

- A new property will be added to the object
- An existing property will be removed from the object (if permitted)
- An existing property will be modified to match the import data
- An existing property will remain the same, although the import data contains a different value, since the property cannot be modified
You should view the details and accept these changes, if appropriate.

19.3.3 Using the Details window

The Details of Import Definition and Target Definition window displays the properties of an object, and for each property, its value in the import file (in the Source column), and the target system (in the Existing target column). It also shows the value (in the Resulting target column) that will be imported (Figure 19-14).

![Figure 19-14  Details of Import Definition and Target Definition window](image)

Each property is assigned one of three states:

1. **Same** indicates that the source, existing target, and resulting target are all identical.

2. **New** indicates that there is no value in the existing target, and that the source and resulting target are identical.

3. **Different** is assigned to any other situation. Logically, either:
   
a. The source and existing target are different. The resulting target will be set to Source if the property can be modified, otherwise it will be set to Existing target.
b. The existing target is empty (this is a new property), but the source and resulting target differ. This can only occur for properties which are validated, as listed in Table 17-1.

You should review the properties which are different and click **Accept** or **Do not import**, as appropriate. Some properties actually reference a property group, which can be viewed by clicking the underlined **Source** value (Figure 17-20). If the group contains a property which is in the different state, the entire group will be marked different, thus it will appear even when the **Only show properties that have differences** radio button is selected.

![Figure 19-15 Details window with property group](image)

### 19.3.4 Completing the import

Once you have checked all of the **Different** and **New Different** objects in the Import Preprocessor Results window, you can click **Continue** to move on to the Confirm Import Selection window (Figure 19-16). Note that you can no longer move back to the previous window, so if you decide to make changes to the import data, you must click **Cancel** here and begin the import process again.
After you choose the files into your target system in the Import Preprocessor Results window, you can choose how you want to handle any errors during the import process.

From the Confirm Import Selection window (Figure 17-21), you can review the list of objects that you selected to import and decide how to proceed.

1. Choose whether to continue importing the objects if an error occurs:
   - **Log error and abort.** If an error occurs, the system adds the error to a log file and cancels the import action.
   - **Log error and continue.** If an error occurs, the system adds the error to a log file and continues importing objects.

2. Select how you want to proceed:
   - **Import.** Begin the import process.
   - **Save.** Save the selections that you made, including which objects to import and properties you have modified, to a file so you can import later. The Save Import File As window displays, from which you can select the directory and specify a file name.
   - **Cancel.** Closes the window without saving any information.
Finally, select a logging option and click **Import** to begin importing the objects. Following completion, you will find the log file, `userid.cmbxml.log`, in the system administration directory. Its contents include a list of the objects which were imported along with a record of any errors which occurred, similar to the following example:

```
XML Import at 12/19/05 6:00 PM
Access control list data: PublicReadACL
Item type: MyItemType
Imported 2 of 2 items.
```

### 19.3.5 Importing a process from XML text

Open the graphical process builder by opening an existing process or defining a new one. The process you import will be included in this new or existing process.

You can import a process that you previously exported as XML text from the graphical process builder. The primary reason to use this functionality is to move built and verified processes from a test system to a separate production system.

**Restriction:** You cannot use this functionality to import a process that you exported as XML in the System Administration Client window (for example, by right-clicking a process and clicking **Export All as XML**); this XML text import function works only with files that you previously export as XML text from the graphical builder. To import an XML file that you exported as XML in the System Administration Client window, you must click **Tools → Import from XML** in the System Administration Client window.
To import an XML text file that you exported from the graphical builder, proceed as follows:

1. Within the graphical process builder, click **File → Import XML text** (Figure 19-17).

![Figure 19-17](image-url)  
*Figure 19-17  Select the Import XML text from graphical process builder*
2. Select the XML file that you want to import (Figure 19-18).

![Figure 19-18 Selecting the XML file](image)

3. Click Select XML File. If you selected a file with the same name as a process that currently exists in the Library Server, a warning displays. If you want to save such an imported diagram, consider using File → Save As to give it a different name (Figure 19-19).

![Figure 19-19 Workflow process imported from XML file](image)
4. Verify the process to determine whether any required objects are missing from this system. Importing an XML text process into the graphical builder does not automatically create the necessary document routing objects (for example, work nodes) for the process.

5. Create necessary document routing objects. You can either create those objects manually, or use the XML export functionality from the System Administration Client window to export them and then import them to the new system.

6. Re-verify the process as necessary.

7. Save and close the verified process.

19.4 Importing and exporting metadata using XML services APIs

You can import and export Content Manager metadata using Content Manager System Administration Client or use APIs directly.

There are two types of metadata you can import:

- **Administrative objects:**
  Users, user groups, Resource Manager configuration, document routing definitions, ACL privileges, and the WebSphere Information Integrator for Content search template and server configuration. This uses an XML schema called cmadmin.xsd (located in IBMCMROOT/config/) to define the XML files containing Content Manager Version 8 administration objects.

- **Data model objects:**
  The structure of item types, component types, and WebSphere Information Integrator for Content entities. These objects are stored as XSD files known as storage schemas. Each storage schema imports a common file named cmdatamodel.xsd (located in IBMCMROOT/config).

By representing Content Manager metadata as intermediate files, you can program a number of scenarios:

- Customizing an application to administer and update data in Content Manager through the XML interface
- Transferring metadata from one Content Manager system to another Content Manager system (taking into consideration any data conflicts)
Transferring entities, search templates and server configuration from one WebSphere Information Integrator for Content system to another WebSphere Information Integrator for Content system (taking into consideration any data conflicts)

These scenarios become important in typical business situations such as:

- During the deployment of your content management application, transferring metadata from a test system to a production system.
- During the extension of an application or addition of a new Content Manager production system, transferring specific objects between development, test, and production systems. In this case, the existing data in the target system is updated.
- During troubleshooting of a production system, transferring specific objects from a production system to a test system in order to diagnose the problem.

The XML metadata service class, `DKXMLSysAdminService`, contains three new Version 8.3 methods for importing and exporting Content Manager metadata: `list()`, `ingest()`, and `extract()`. The latter two methods import and export storage schemas (XSD files) for data model objects, and XML files for administration objects (using the pre-defined `cmadmin.xsd` schema).

The `ingest()` and `extract()` methods support the following formats:

- **XML input formats**:
  - `DKXMLStreamObjectDefs`: input stream
  - `DKXMLDOMObjectDefs`: DOM

- **XML output formats**:
  - `DKXMLDOMObjectDefs`: DOM

`DKXMLDOMObjectDefs` has a method to convert the DOM object into an output stream.

Additionally, `DKXMLSysAdminService` supports the following `ingest()` options:

- **`DK_CM_XML_IMPORT_CONTINUE_ON_ERROR`**
  If you OR this constant, then the `ingest()` method imports as many object definitions as possible. If you neither specify nor OR this constant, then the import process aborts on any error

- **`DK_CM_XML_IMPORT_CREATE_UPDATE`**
  If you specify this constant, then the `ingest()` method replaces objects that already exist. If you do not specify this constant, then the import fails with an error for any object definitions that already exist
The **DKXMLDOMObjectDefs** class provides two methods, `getSysAdminDefs()` and `getDataModelDefs()`, to retrieve the exported data model objects (in XML Schema format) and administrative objects (in XML format) separately. The **DKXMLExportList** class can specify which XML objects to export (for any objects that require other objects to work).

### 19.4.1 Importing and exporting administration objects as XML

You can import and export administration objects using Content Manager System Administration Client or APIs.

The XML metadata service class, **DKXMLSysAdminService**, contains two new Version 8.3 methods for importing and exporting Content Manager metadata: `ingest()` and `extract()`. These methods import and export XML files for administration objects that conform to the `cmadmin.xsd` schema.

The **DKXMLDOMObjectDefs** class provides two methods, `getSysAdminDefs()` and `getDataModelDefs()`, to retrieve the exported data model objects (in XML Schema format) and administrative objects (in XML format) separately. The **DKXMLExportList** class can specify which XML objects to export.

The following Content Manager system administration objects (represented by constants in the `com.ibm.mm.sdk.common.DKConstant` class) can be converted to and from XML:

- Administrative domain
- Privilege definitions
- Privilege groups
- Privilege sets
- Users
- User Groups
- ACLs
- Library server configuration
- Library server language definition
- Link type
- MIME type
- Semantic type
- XDO class
- Resource manager objects
- Document routing objects

The following WebSphere Information Integrator for Content system administration objects can be converted to and from XML:

- Server definition
- Federated entities
- Search templates
The exported schema for a given data model object is semantically equivalent to the imported schema that creates it. That is, by exporting and importing an object from one system to another system, all of the object properties should be the same to the original exported one. However, the exported schema document and the imported schema document might differ in syntax. This is because of the many different ways for XML Schema to represent the same information.

Example 19-1 creates a new user (Joshua) and a new group (XMLDev) in the Content Manager server:

Example 19-1   XML example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CMSystemAdminDefinitions
xmlns="http://www.ibm.com/xmlns/db2/cm/api/1.0/schema"
xmlns:cm="http://www.ibm.com/xmlns/db2/cm/api/1.0/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <user name="JOSHUA" adminDomainName="SuperDomain"
    defaultRM="RMDB"
    defaultSMSColl="CBR.CLLCT001" description="Regular user"
    passwordExpiration="0" userACL="PublicReadACL"
    userGrantPrivilegeSet="ClientUserReadOnly"
    userPrivilegeSet="AllPrivs">
    <userGroup name="XMLDEV"/>
  </user>
  <userGroup adminDomainName="PublicDomain"
    description="XML Development" name="XMLDEV"/>
  <groupData groupName="XMLDEV">
    <user userName="JOSHUA"/>
  </groupData>
</CMSystemAdminDefinitions>
```

19.4.2 Importing and exporting Content Manager data model objects as XML schema files (XSD)

You can import and export Content Manager data model objects using Content Manager System Administration Client or APIs.

The XML metadata service class, **DKXMLSysAdminService**, contains two new Version 8.3 methods for importing and exporting Content Manager metadata: **ingest()** and **extract()**. These methods import and export storage schemas (XSD files) for data model objects.

The **DKXMLDOMObjectDefs** class provides two methods, **getSysAdminDefs()** and **getDataModelDefs()**, to retrieve the exported data model objects (in XML Schema format) and administrative objects (in XML format) separately. The **DKXMLExportList** class can specify which XML objects to export.
Generally, the following rules apply when your storage schema is imported into Content Manager:

- A root element declaration (for example, an insurance policy) is mapped to an `XYZ_InsPolicy` item type
  ```xml
  <xsd:element name="XYZ_InsPolicy">
  </xsd:element>
  ```

- A child element declaration (for example, a vehicle identification number) is mapped to an `XYZ_VIN` component type under the corresponding parent component type (in this example, the `XYZ_InsPolicy` root component type)
  ```xml
  <xsd:element maxOccurs="unbounded" minOccurs="0" name="XYZ_VIN">
  </xsd:element>
  ```

- An attribute inside of an element declaration is mapped to an attribute in the corresponding component (for example, a policy's ID number attribute maps to an `XYZ_PolicyNum` attribute in the policy item type)
  ```xml
  <xsd:attribute name="XYZ_PolicyNum">
  </xsd:attribute>
  ```

- In accordance with the SQL/XML standard for mapping SQL identifiers to XML names, the XML schema converter automatically escapes special characters with the unicode equivalent (in the form of _xYYYY_, where YYYY represents the Unicode string). For example:
  
  - Elements and attribute names cannot start with the letters XML. Therefore, if an item type is named `XMLDocument`, then its new name becomes: `_x0058_MLDocument`
  
  - Elements and attributes names cannot contain spaces. Therefore, if a federated entity is named `Project Entity`, then its new name becomes: `Project_x0020_Entity`

Example 19-2 shows a sample storage schema (XSD) snippet for the XYZ Insurance policy item type:

**Example 19-2  Sample Storage Schema (XSD)**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:cm="http://www.ibm.com/xmlns/db2/cm/api/1.0/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <xsd:import namespace="http://www.ibm.com/xmlns/db2/cm/api/1.0/schema"
    schemaLocation="cmdatamodel.xsd"/>
  <xsd:attribute name="XYZ_VIN">
    <xsd:annotation>
      <xsd:appinfo>
        <cm:description value="Vehicle Identification Number (Content Manager Sample Attribute)" xsi:lang="ENU"/>
      </xsd:appinfo>
    </xsd:annotation>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:length value="17"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="XYZ_InsrdLName">
    ...
  </xsd:attribute>
  <xsd:attribute name="XYZ_InsrdFName">
    ...
  </xsd:attribute>
  <xsd:attribute name="XYZ_ZIPCode">
    ...
  </xsd:attribute>
</xsd:schema>
```
The exported schema for a given data model object is semantically equivalent to the imported schema that creates it. That is, by exporting and importing an object from one system to another system, all of the imported object's properties remain the same as those of the original exported one. However, the exported schema document and the imported schema document might differ in syntax because of the many different ways for XML schema to represent the same information.

Content Manager defines the storage schema using the following steps:

1. Content Manager attempts to map any available construct or feature in the XML schema to a Content Manager data model concept

2. If no Content Manager concept directly corresponds to the XML schema, then the concept is instead represented by a comment (also known as an XML annotation)

3. Content Manager instances (for example, item-level ACLs and semantic types) are represented as XML elements (in the Content Manager namespace) imported from the cmdatamodel.xsd file
19.4.3 Unsupported XML types in the Content Manager storage schemas

The Content Manager storage schemas do not support the following primitive data types:

- string (has to be associated with either length properties)
- boolean
- duration
- gYearMonth
- gYear
- gMonthDay
- gDay
- gMonth
- hexBinary (only support base64Binary)
- QName
- NOTATION

The Content Manager storage schemas do not support the following derived data types:

- normalizedString
- token
- language
- NMTOKEN
- NMTOKENS
- Name
- NCName
- ID
- IDREF
- IDREFS
- ENTITY
- ENTITIES
- nonPositiveInteger
- negativeInteger
- long
- byte
- nonNegativeInteger
- unsignedLong
- unsignedInt
- unsignedShort
- unsignedByte
- positiveInteger
19.4.4 Constraints for converting to Content Manager storage schemas

The following constraints also apply to the Content Manager storage schema conversion:

- The `minLength` and `maxLength` attribute values (if specified) must be the same, which describes the `DK_CM_CHAR` data type.
- An element name, which maps to component name, cannot be used in different symbol space.
- Any attribute with the same name must have the same basic type.

**Restriction:** Some properties of the attributes with the same name can be different, such as `maxInclusive` and `minInclusive`.

- The `xsi:type` and `xsi:nil` attributes are not supported in the instance document.
- No recursive type definition is allowed. For example, the following definition is not allowed (Example 19-3):

```
Example 19-3  No recursive definitions are allowed
<xs:element name="Section">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Section" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="title" use="required"/>
    <xs:attribute ref="content" use="required"/>
  </xs:complexType>
</xs:element>
```

19.4.5 Importing and exporting Content Manager data instance objects as XML

The XML instance service class, `DKXMLDataInstanceService`, contains two new Version 8.3 methods for importing and exporting XML items: `ingest()` and `extract()`. These methods take the XML file that conforms to the storage schema (described in 17.4.2 “Importing and exporting Content Manager data model objects as XML schema files (XSD)”, which can be exported through the extract API or the System Administration Client, on any item type) to structure the data instance objects (such as items and documents with parts). Their input and return parameters work similarly to the old Version 8.2 methods, `toXML()` and `fromXML()`.
The `ingest()` and `extract()` methods support the following formats:

- **XML item input formats**
  - `DKXMLDOMItem`: document object model (DOM)
  - `DKXMLStreamItem`: input stream (processed using SAX)
  - `DKXMLStringItem`: string

- **XML item output formats**
  - `DKXMLDOMItem`: DOM (default)
  - `DKXMLStringItem`: string

`DKXMLDOMItem` features a method that can convert an XML item from DOM format to Input stream format.

Example 19-4 shows a sample item instance that conforms to the XYZ Insurance policy storage schema:

**Example 19-4  Sample item instance for storage schema**

```xml
<Item><ItemXML>
  <XYZ_InsPolicy XYZ_Street="532 Camino Viejo"
    XYZ_City="Marina" XYZ_State="CA" XYZ_ZIPCode="90546"
    XYZ_PolicyNum="57904965371" xmlns="">
    <XYZ_Insured XYZ_InsrdFName="Edward" XYZ_InsrdLName="Smith" />
    <XYZ_Insured XYZ_InsrdFName="Jennifer" XYZ_InsrdLName="Smith" />
    <XYZ_VIN XYZ_VIN="ICLA44P5KL9876543" />
    <ICMBASE><resourceObject MIMEType="image/tiff"
      xmlns="http://www.ibm.com/xmlns/db2/cm/api/1.0/schema"><label
        name="policyForm" /></resourceObject></ICMBASE>
  </XYZ_InsPolicy>
</ItemXML></Item>
```

### 19.4.6 Exporting Content Manager DDO items as XML items

The `extract()` method in `DKXMLDataInstanceService` converts a DDO (including all child component DDOs, links, and references) into a `DKXMLItem` object. This `DKXMLItem` object contains the following data:

- XML document that represents the item version, including all child components
- Properties, including system attributes, resource attributes, and links information
- Any binary resource part content

**Tip:** Pass in the `DKConstant.DK_CM_XML_EMBED_UNIQUE_IDENTIFIER` to TRUE in order to include the resource content's part number.
The `extract()` method accepts the DDO (and various options) as the input parameters. The various options include:

- **Which XML format to export the item and its properties:**
  - `DKXMLDOMItem`: `DKConstant.DK_CM_XML_DOM_FORMAT` (this is the default)
  - `DKXMLStreamItem`: `DKConstant.DK_CM_XML_RESOURCE_STREAM_FORMAT`
  - `DKXMLStringItem`: `DKConstant.DK_CM_XML_DOM_FORMAT`

- **Which output format to export the resource content as (URL or input stream).** URL is the default. If you select input stream, then the system generates unique labels to identify each resource. These labels can be found in the resource properties.

- **Whether to include the PID and part number with the XML item.**

- **Whether to export the item's properties as a separate XML document.** Use this option to exclude all proprietary Content Manager information from the XML item.

Example 19-5 inputs a `ddo` item and returns it as an `xmlobj` XML document (in DOM format with the PID embedded in it); returns system and resource properties in a separate file, and returns resource content as an input stream.

**Example 19-5   Export DDO item example**

```java
//create an instance of DKXMLDataInstanceService
DKXMLDataInstanceService instService = new DKXMLDataInstanceService(dsICM);
DKXMLDOMItem xmlObj =
(DKXMLDOMItem) instService.extract(ddo,DKConstant.DK_CM_XML_DOM_FORMAT +
DKConstant.DK_CM_XML_SYSTEM_PROPERTY_REFERENCE +
DKConstant.DK_CM_XML_RESOURCE_PROPERTY_REFERENCE +
DKConstant.DK_CM_XML_RESOURCE_STREAM_FORMAT +
DKConstant.DK_CM_XML_EMBED_UNIQUE_IDENTIFIER);
//get the XML document representing item version
Document xmlDocument = xmlObj.getXMLItem();
//get the XML document with properties
Document propertyDocument = xmlObj.getItemProperties();
//get content labels
Set resLabels = xmlObj.getContentLabels();
//create an iterator
Iterator iter = resLabels.iterator();
//iterate over the set to get labels and resource contents
while (iter.hasNext()) {
    //get the label from the iterator
    String label = (String)iter.next();
    // get resource content as input stream from xml object
    BufferedInputStream inStream = new
        BufferedInputStream(xmlObj.getContentAsStream(label));
}`
19.4.7 Importing XML items as Content Manager DDO items

The `ingest()` method in `DKXMLDataInstanceService` converts a `DKXMLItem` object into a DDO on the Content Manager server. These constructors extract content from an XML document, and create a corresponding DKDDO and any dkXDO associated with it. You can then call the add method on the DDO to add the object into Content Manager.

The new DDO belongs to a Content Manager Version 8 item type or an earlier Content Manager index class and can only be stored in Content Manager.

Importing an XML file allows you to store the original XML file as an XDO, that is, you do not lose the XML in the import process, making the XML itself available for possible future use.

As you import XML content, keep the following facts in mind:
- You can only import into Content Manager or earlier Content Manager.
- XML files containing content for import must conform to the storage schema of the corresponding item type, which you can export through the API described in 17.4.6 “Importing and exporting Content Manager data instance objects as XML” or the System Administration Client.
- XML import and XML export are supported only by the Java APIs.

The `input()` method accepts the following input parameters:
- XML document in a `DKXMLDOMItem`, `DKXMLStreamItem`, or `DKXMLStringItem`. If you input a `DKXMLStreamItem`, then a SAX handler converts the input stream to a DDO object (not DOM).
- A pre-existing DDO to populate the XML data (optional).
- Resource content as a `DKXMLItem` object in input stream format. Using the `DKXMLItem.setContentAsStream()` method, you can create unique labels for the resource properties for `ingest()` to interpret.
- Properties such as system attributes, resource attributes, and links information. You can either embed them in the original XML document, or import them as a separate XML document from the original XML document that describes the item. You can either provide this document through the `setItemProperties()` method or in the constructor.

Example 19-6 inputs both an XML item `XMLFile` and its properties (both system and resource in a separate file `XMLProperties`) as input streams; and returns a Content Manager `ddo`.
Example 19-6  Importing XML item as Content Manager ddo

```
//create file stream for XML document representing item version
FileInputStream xmlDocument = new FileInputStream(XMLFile);
//create file stream for XML document representing properties
//properties include system properties, resource properties
FileInputStream properties = new FileInputStream(XMLProperties);
//Create an instance of DKXMLStreamItem
DKXMLStreamItem xmlItem = new DKXMLStreamItem(XMLFile, XMLProperties);
//set value for resource content label
String contentLabel = "AAA";
//Set resource content into xmlItem
xmlItem.setContentAsStream(contentLabel, contentStream);
//create an instance of DKXMLDataInstanceService
DKXMLDataInstanceService instService = new DKXMLDataInstanceService(dsICM);
//call ingest on instance service
DKDDO ddo = (DKDDO) instService.ingest(xmlItem, options);
```

19.4.8 Importing and exporting XML object dependencies

Scenarios can occur where data model and administrative objects require the existence of other definitions (dependency objects) in the server. For example, a user must be defined before you can define a user group for it.

By default, the `extract()` method only exports the object and no dependency objects. In order to prevent problems from missing dependencies, you can specify one of the following options for exporting objects to XML:

- **DK_CM_XML_EXPORT_PREREQUISITE**
  Exports all dependency objects with the object

- **DK_CM_XML_EXPORT_DM_ONLY_PREREQUISITE**
  Exports only data model dependency objects

- **DK_CM_XML_EXPORT_SA_ONLY_PREREQUISITE**
  Exports only administrative dependency objects (including authorization, authentication, and Library Server configuration)

- **DK_CM_XML_EXPORT_RM_ONLY_PREREQUISITE**
  Exports only Resource Manager configuration (in the Library Server side) dependency objects

- **DK_CM_XML_EXPORT_DR_ONLY_PREREQUISITE**
  Exports only document routing dependency objects.

During an import, if the dependency objects do not exist in the target system, an exception is logged or the process is aborted (depending on the error handling option set.)
19.4.9 Extracting content from different XML sources

The **DKDDO** methods can extract content from a variety of XML sources, including standard input, files, buffers, and Web addresses (URLs). Call the **DKDDO** methods to extract content from your XML source and to initiate the import process.

Some examples of each XML source are shown in Example 19-7, Example 19-8, and Example 19-9.

**Example 19-7  XML from a file (Java)**

```java
xmlSource = new DKNVPair("FILE", "dlsamp01.xml");
```

**Example 19-8  XML from a buffer (Java)**

```java
File file = new File("dlsamp01.xml");
int fileSize = (int) file.length();
byte[] data = new byte[fileSize];
DataInputStream dis = new DataInputStream(new FileInputStream(file));
dis.readFully(data);
String strBuffer = new String(data);
DKNVPair xmlSource = new DKNVPair("BUFFER", strBuffer);
int importOptions=DK_CM_XML_VALIDATION;
```

**Example 19-9  XML from a Web address (Java)**

```java
xmlSource = new DKNVPair("URL", "file:///d://myxml//dlsamp01.xml");
// replace file:///d:// with http://www.webaddress.com/ for URL
int importOptions=0;
```
Performance tuning

In this chapter, we provide the basics of what to look for when trying to fine-tune your Content Manager implementation and improve system performance. We discuss how to address performance for each of the Content Manager components, as well as the supporting infrastructure components such as the WebSphere Application Servers, DB2 Universal Database, and Tivoli Storage Manager.
20.1 Performance tuning basics

Performance requirements should be kept in mind as early as the planning stage of a Content Manager implementation. Several factors, such as system configuration, number of servers, type of servers, number of users, and peak volume usage affect system performance. For a set of best practices in planning for a highly performing system, refer to 11.7, “Planning for performance” on page 295. Regardless of how well you plan and implement the system, there will be times in which your system will require tuning. Some of the reasons include changes in workload characteristics, upgrade to a new Content Manager version, or poor system performance.

To ensure that your system is performing at the best possible level, you should:

- Perform routine system maintenance (refer to Chapter 18, “Maintenance” on page 471).
- Periodically monitor the system performance (see 20.2, “Performance monitoring” on page 546).
- Periodically tune the system for performance and avoidance of potential crisis (see 20.3, “Performance tuning” on page 549).

In this section, we cover what you should know prior to monitoring and tuning your system for performance:

- Understanding performance goals
- General performance tuning guidelines
- Content Manager components

Note: Much of this chapter is an extract from the IBM Redbook *Performance Tuning for Content Manager*, SG24-6949. We strongly recommend reading this redbook for detailed Content Manager performance tuning information.

20.1.1 Understanding performance goals

Whether you are in a proactive or reactive mode, you should first understand what your performance goals are. Not understanding the business objectives would lead to tuning for the sake of tuning. Business Volume Metrics (BVM) dictates how the system should perform. Most BVMs are measured in terms of response time and throughput.

Response time is the elapsed time between when a request is submitted and when the response from that request is returned. For example, you need to retrieve a document within a second. This one second becomes your performance goal for document retrieval response time.
Throughput is a measure of the amount of work over a period of time. A typical example is to complete 5 transactions per second (TPS) for the system.

Another important measure of a highly available system is how well you have planned for system failure. Do you have redundant servers and have effective load balancing techniques? Can the backup server handle as much load as the original system? Sometimes crisis situations arise because of system failure resulting in degraded performance or a total halt to business.

When tuning for performance, the basic bottlenecks to be resolved are CPU, memory, disks, and network. This is where capacity planning plays a vital role for a well planned system. If you use a performance model that has all these bottlenecks represented, it will help you with projections for upgrading this vital resource and will help to make your system more scalable. In summary, the basic goals for performance tuning, whether reactive or proactive, are:

- Faster response time
- Increased throughput
- Increased system availability
- Decreased bottlenecks

**Note:** It is very important that the goals are reasonable and measurable.

### 20.1.2 General performance tuning guidelines

These are some guidelines you should follow for successful performance tuning:

- Establish *quantitative, measurable, and realistic* objectives.
- Understand and consider the entire system to give a clear picture of the components and sub-components of your solution.
- Change one parameter at a time to understand and pinpoint bottlenecks.
- Measure and reconfigure by each component one at a time: hardware, operations system, Library Server, Resource Manager, database, and WebSphere Application Server.
- Consider design and re-design. Recognize that some elements may have to be redesigned due to change in usage metrics.
- Remember the law of diminishing returns. Stop when the outcome is already acceptable, because changes after a while may not produce drastic returns.
- Recognize performance tuning limitations. A case in point is adding more CPUs, memory, or disks.
- Understand the configuration choices and trade-offs.
20.1.3 Content Manager components

Remember that the Content Manager system consists of these components:

- Library Server
- Resource Manager
- Windows Client, eClient, and/or customer client

Each of the Content Manager components uses other underlying software or infrastructure elements to work. For example:

- Library Server requires DB2 UDB or Oracle.
- Resource Manager requires WebSphere Application Server and may optionally use TSM services.
- Content Manager clients may require Content Manager Java or C++ APIs.

Many of these components may be run on a mix of platforms such as AIX, Solaris, Windows, and z/OS.

Given that Content Manager utilizes many components and underlying infrastructure elements, it is important to understand that monitoring and fine tuning Content Manager means monitoring and fine tuning one or more of these individual components of the Content Manager solution package.

20.2 Performance monitoring

Periodic monitoring of system metrics is vital to the health of your Content Manager solution. Monitoring is essential for the performance tuning and improvement process. It also helps to be proactive and identify potential bottlenecks. As we discussed earlier in this chapter, your Content Manager solution includes multiple components and supporting infrastructure.

There is no single tool that can monitor the health of the entire system. You should use specialized tools to monitor each component or sub-component. In this section, we provide a high level overview of some of the tools that are available. For detailed literature on these tools and usage, please refer to the individual reference publications. Also remember that the objective of monitoring is to sample data at certain intervals (peak and non-peak hours), not to run regularly and be intrusive to the production system.

**Monitoring Windows**

The Task Manager is the key tool in monitoring Windows resources. There are three tabs: Applications, Processes, and Performance. The Performance tab gives a graphical representation and detailed information about the total CPU
usage and Memory usage history on the machine. CPU information includes handles, threads, and processes currently being executed. Memory usage information gives physical, kernel, and commit memory. The Processes tab in Task Manager gives the CPU and memory usage information broken down by each active process on the machine.

**Monitoring AIX**

The AIX platform has several command line tools that give information about system resources. Some of the commands are listed below:

- To monitor CPU usage: `sar -u`
- To monitor memory usage: `vmstat`
- To monitor disk I/O: `iostat`
- To monitor Network I/O: `netpmon`

The `topas` command is a recommended tool that you can use to display information by `sar`, `iostat`, and `vmstat` commands. The tool reports local system statistics such as CPU use, CPU events and queues, memory and paging use, disk performance, network performance, and NFS statistics. It also reports the top hot processes of the system. All information displayed by `topas` is real time.

**Monitoring DB2**

DB2 provides two types of monitoring: snapshot and event monitoring. Snapshot monitoring is used for obtaining database relevant statistics at a specific point in time and at various levels of DB2 objects. The snapshot levels are: database manager, database, application, and buffer pool. Furthermore, for each level you can use up to six switches to see the maximum information. The switches are: Sort, Lock, Table, Buffer pool, UOW, and Statement.

Event monitoring is another type of monitoring provided by DB2 that gives database statistics collected over a period of time. The main events about which statistics are gathered include: Database, Tables, Deadlocks, Table spaces, Buffer pools, Connections, Statements, and Transactions. After you have collected event statistics over a period of time, these monitors can be turned off. Their output can be written to file and analyzed using tools available with DB2: `db2evmon`, the text-based tool, and the Event Monitor GUI tool. The monitor files can also be off-loaded to an external share for storage and later analysis.

DB2 also provides the Health Center, available in DB2 V8, a daemon program that continuously runs to monitor the health of DB2 UDB. Some monitored health indicators are free memory, table space containers, and logging storage. You can define low and high thresholds for these indicators. When an indicator's value falls outside this zone, Health Monitor generates an alert.

Otherwise, we supply a tool named DB2 Performance Expert V2.2 for multiplatforms. This tool will help to monitor and analyze the performance of the
Content Manager DB2 database. For more detailed information, refer to *Monitoring IBM DB2 Content Manager V8.3 Enterprise Edition with DB2 Performance Expert V2.2 for Multiplatforms*.

**Monitoring WebSphere Application Server**

The WebSphere Application Server provides the Performance Monitoring Infrastructure (PMI), a set of packages and libraries, for collecting, analyzing and displaying data in the WebSphere Application Server run time application components. Data can then be analyzed with any number of available tools such as Tivoli Performance Viewer (TMV), Tivoli Monitoring for Web Infrastructure and Tivoli Monitoring for Transaction Performance.

Tivoli Performance Viewer allows you to view WebSphere Application Server data real time or off line, view data in different graphical forms, and compare data for single resource to an aggregate of resources. Using TMV, you can estimate the load on application servers and the average response time for clients. TMV provides information on application-specific resources such as enterprise beans and servlets; it also gives information on the WebSphere Application Server runtime resources such as the Java Virtual Machine (JVM™).

**Monitoring Content Manager**

There are several trace utilities available to trace the performance of Library Server, Resource Manager, and the Content Manager connector. To enable Library Server performance tracing, the following DB2 script can be used:

```sql
db2 connect to icmnlsdb user icmadmin
db2 update icmstsyscontrol set tracelevel=-8
db2 update icmstsyscontrol set keeptraceopen=1
db2 connect reset
```

The log file will go to a location specified using the System Administration Client, for example, C:\ICMServer.log on Windows.

For Resource Manager, you can use the XML file icmrm_logging.xml in the corresponding $IBMCMROOT\cmgmt directory. The priority value in the XML file can be chosen to be one of several values such as ERROR or WARN. The trace data will be logged in the icmrm.logfile file. By default, this file will be located in the $IBMCMROOT\log directory. We can configure the location by modifying the *File Value* parameter of the icmrm_logging.xml configuration file. To enable a continuous performance trace over a period of time, change the priority value in the XML file to BEGINEND, and stop and restart WebSphere Application Server.

The Content Manager connector can be traced using the file cmblogconfig.properties, in the $IBMCMROOT\cmgmt\connectors\ directory. In that file, the DKLogPriority can be set to one of several values such as ERROR or INFO. The trace data will go into the file specified by DKLogOutputFileName.
To start a trace on the Content Manager connector, edit the file cmblogconfig.properties, change the DKLogPriority value to TRACE, and stop and start the system.

For more information on monitoring Content Manager and trace information, refer to Chapter 21, “Troubleshooting” on page 559.

### 20.3 Performance tuning

We have discussed the monitoring of individual component and sub-components of your Content Manager implementation. Similarly, in this section, we provide a high level view of tuning these components. Much like monitoring, tuning is an exercise that has to be performed periodically to keep your system in top health.

For more detailed information, refer to the IBM Redbook *Performance Tuning for Content Manager*, SG24-6949.

#### 20.3.1 Tuning Windows

As indicated by the Task Manager monitoring tool, there are several resources on Windows that can be tuned. In general, you want to disable unnecessary processes and services and increase memory capacity.

Disabling Windows services that are not needed for Content Manager may reduce system overhead and increase available resources. Services can be disabled using the **Start → Settings → Control Panel → Administrative Tools → Services** applet. Some of the services you can disable include:

- Indexing service
- Computer browser service
- IIS Admin Service

Set the following registry key value to 1 to disable 8.3 short filename creation using `regedit`:

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\FileSystem\NtfsDisable8dot3NameCreation
```

Check if your Windows server™ has enough memory for Content Manager. The rule of thumb for the Library Server is to have 1 GB RAM plus 10 MB RAM for each Windows desktop client plus 10 MB RAM for every 20 Web-based clients.

For more detailed information on hardware requirements, refer to the Content Manager manual, *Planning and Installing Your Content Management System*. 
20.3.2 Tuning AIX

To maximize AIX performance, following are some changes you can make to the operating system configuration:

- Adjust maximum number of PROCESSES allowed per user from the default 1024 to a suitable value, depending on the number of concurrent users and number of processes running on the system. For example, you may have to increase the value to 8192.
- Define your logical volumes and file system using Journaled File System 2 (JFS2) and Enhanced Journal File System (EJFS) to manage high volume of documents.
- Check the user process limits defined in the /etc/security/limits file. Verify that all values used for the user db2instl user are defined in this file. If not, edit or use the ulimit command to add or change the values.

20.3.3 Tuning DB2

DB2 is at the core of the Content Manager system functions and performance. There are numerous parameters and tuning guidelines for DB2. In this section, we highlight some of the key DB2 performance tuning guidelines.

20.3.4 Use multiple disks

To maximize I/O concurrency and improve scalability, have multiple disks and use the following separation policies:

- Separate log files from database and put them on separate disks.
- Separate instances (if AIX).
- Separate databases.
- Separate tablespaces across multiple physical disks.

20.3.5 Customize Content Manager database installation

There is another way to execute the separation policies to spread DB2 components. You can do so by customizing the database installation scripts, also known as data definition language (DDL) or schema scripts. The DDL files are located in the <IBMCMROOT>/config directory. Changes require careful planning and should be done by an experienced database administrator.

20.3.6 Separate database instances

On an AIX system, the default installation is to use one database instance for both the Library Server and Resource Manager databases. One disadvantage of this approach is that all databases on the single instance have to share the
maximum of 1.7 GB memory for database buffer pools. Thus we recommend that you create one database instance per database.

20.3.7 Create attribute indexes

During installation, the Library Server and Resource Manager database tables are created with appropriate indexes to optimize database operations. If you create a new item attribute, indexes are not automatically created. Creating indexes for these attributes that are frequently used in queries will improve system performance.

20.3.8 Routine runstats/rebind

As an administrative process, run runstats and rebind periodically to maintain database health as described in 18.2, “Optimizing server databases” on page 472. This keeps execution plans for your database up to date.

20.3.9 Tuning WebSphere Application Server

Resource Manager is a WebSphere Application Server application, and hence, tuning WebSphere Application Server performance directly influences Content Manager performance. By periodically monitoring for system resource and configuring hardware on which WebSphere Application Server is installed, you can avoid bottlenecks. Check to make sure you have a high speed processor, sufficient system memory (at least 256 MB for each processor) and reasonable network latency (at least 100 MB on 10/100 Ethernet).

Follow the operating system tuning guidelines we discussed earlier. On Windows, you can increase the TCP wait time to a value of 30 seconds for the following registry key parameter:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters
```

Create a REG_DWORD named TcpTimedWaitDelay and set its value to 30.

Another registry value that can be changed beneficially is to increase maximum ports available for client connections. The parameter is MaxUserPort and it is in the same registry key path as the TcpTimedWaitDelay parameter mentioned above. Increase the MaxUserPort value to 65534.

Your Web server parameter values can also make a difference. Make sure they are consistent with your application and throughput requirements. The MaxClients parameter in IBM HTTP Server has a default value of 150. You may need to change it depending on how CPU intensive or database intensive your application is.
WebSphere Application Server contains application servers that have EJB and Web containers. Improving the operating system process priority of the application server can improve performance. Make sure changing to a higher priority does not affect other system parameters. Some other WebSphere Application Server parameters you may have to tweak are Web container maximum thread pool, MaxKeepAliveConnection value for the HTTP Transport property, and MaxKeepAliveRequests value for the HTTP Transport property.

Java Virtual Machine (JVM) settings affect WebSphere Application Server performance. Check the minimum and maximum heap sizes. As an example, set the maximum to 256 MB for a system with 2GB memory, 512 MB for a larger system.

### 20.3.10 Tuning Content Manager

We have seen that tuning Content Manager involves tuning the operating system, DB2, WebSphere Application Server, and optionally TSM. We have highlighted many of the guidelines in earlier sections. Here are some more tuning guidelines related to the Content Manager components.

#### Library Server

Tuning Library Server is essentially tuning DB2. We covered DB2 tuning earlier. Some more DB2 database manager parameters that will help in tuning are:

- `numdb`
- `maxagents`
- `udf_mem_sz`
- `java_heap_sz`
- `mon_heap_sz`

The database tuning parameters are applicable at the database instance level. Other potential parameters to consider tuning DB2 databases are:

- `dbhheap`
- `logbufsz`
- `logheap`
- `stmtheap`
- `maxfilop`
- `logfilesz`
- `logprimary`
- `logsecond`

Most of these parameters have default values assigned during the Library Server DB2 database installation. It is recommended to start with these values and then fine-tune, based on detailed analysis of system performance.
**Resource Manager**

Most of the Library Server DB2 tuning guidelines apply to the Resource Manager database. WebSphere Application Server tuning is also the key to tuning Resource Manager for performance. Refer to 20.3.9, “Tuning WebSphere Application Server” on page 551 for detail information.

Another effective way to improve Resource Manager performance is to utilize the *LAN Cache* option. LAN Cache enables the local Resource Manager to retrieve the requested object from a remote Resource Manager and store it in its local staging directory. For subsequent client requests for the object, the local Resource Manager retrieves the cached objects from its staging directory instead of accessing the remote server. You can configure to purge the cached objects based on least frequently used objects. Defining the size of the staging directory and the maximum size of the cached objects may also improve performance. These values are depended on your specific document sizes. Use of LAN Cache is recommended when the cache hit ratio is high, you want to reduce the network bandwidth requirements, and reduce load on remote Resource Managers.

*Replication* is another key configuration that helps to maximize your Resource Manager performance. Since replicator is multi-threaded, you can increase it or decrease it depending on the workload and throughput requirements. If you are running replication during off-peak times, increasing the threads increases the replication throughput. If you run replication during regular hours, this may affect overall system scalability. You need to find, by trial and error, an optimal setting for replicator threads to run, so that your ingest rate for workload matches the replication rate.

Migrating document collections from one storage class (disk) to another (for example, tape) reduces storage costs but may impact system performance, since retrieval from tape is slower than from disk. When documents are marked for deletion, an asynchronous process deletes it from the Library Server, Resource Manager, and physically deletes it from the disk. Asynchronously managing the migration process leads to better performance.

### 20.4 DB2 Performance Expert for Content Manager

IBM DB2 Performance Expert for Multiplatforms, a new tool for monitoring Content Manager, makes the job of optimizing performance easier.

DB2 Performance Expert helps optimize the performance and availability of the Content Manager environment, monitoring your system from one control point and sending immediate alerts of problems.
The Performance Expert monitors the Content Manager Library Server and Resource Manager databases and operating system in real-time. It creates detailed reports about the operating system, the Library Server and the Resource Manager databases including the SQL statements executed in the Content Manager databases, and status of buffer pools, table spaces and tables. It also warns of conditions such as locking conflicts and deadlocks, long running SQL statements, memory shortages and file system full conditions.

**Thorough monitoring to prevent problems**
Performance Expert includes predefined system health views specific to Content Manager. Administrators can also define their own customized data views for specific counters and parameters that they need to monitor.

Performance Expert provides proactive out-of-criteria checking and notification and ships with a predefined threshold set for exception processing for Content Manager. Customers can also define a specialized exception threshold set.

These features allow Content Manager administrators to fix problems before they visibly impact user response or system performance.

**Plan ahead with the Performance Warehouse**
Performance Expert includes a Performance Warehouse that stores information about Content Manager's databases and host operating system. The Performance Warehouse enables fact-based tuning of the Content Manager environment. Now the administrator can truly anticipate future requirements:

- Use real data to pinpoint the cause of user reported performance issues.
- Collect performance trace data and generate reports.
- Use aggregated long-term historical data to perform trend analysis and capacity planning.

**Extending the capabilities of Content Manager systems**
20.5 New features for CM C++ API performance

**Important:** The new features that are described in this section will be supported in Content Manager V8.3 Fix Pack 2 or higher.

DB2 Content Manager V8.3 Fix Pack 2 introduces three new features for the Content Manager C++ API: Global Cache, Datastore Pool, and Database Connection Pool. Applications that use the Content Manager C++ API can use these features to improve their response times and scalability. Each feature can be configured separately, but each complements the others and together provide the maximum benefit. These features are especially beneficial for Document Manager mid-tier servers, and custom applications such as Microsoft .NET and DCOM applications where the Content Manager C++ API is running in a mid-tier server environment.

- **Which types of applications will benefit most from these features?**
  Applications in which the threads of a mid-tier server process repeat the patterns of:
  a. Connecting to CM
  b. Performing an operation
  c. Disconnecting from CM

  For these applications, we see the most benefit, but all Content Manager C++ API applications should see some benefit.

- **What operations will benefit?**
  All operations will benefit, including both run-time (CRUD and query) and administration/definition time operations.

- **How does the CM C++ API Global Cache work?**
  Within a process, certain types of Content Manager C++ API objects are retained in memory in a global area that is accessible to all datastores in that process connecting to a Library Server. The benefit is that the Content Manager C++ API does not have to communicate repeatedly with the Library Server to manifest these objects since they have already been manifested and stored in the global cache. This improves response times, and also scalability of both the mid-tier server and the Library Server.
How does the Content Manager C++ API Datastore Pool work?
Within a process, threads which need to access the Content Manager server can obtain connected and logged-on DKDatastore objects from a pool, returning them to the pool when they are done. This can reduce response times on the mid-tier server, and also improve the scalability of the system by reducing the number of connections and calls to the Library Server. In combination with the Content Manager C++ API Global Cache, the Datastore Pool further improves the system scalability since each datastore caches additional definition and administration objects that are not in the global cache, through the life time of the datastore.

How does the Content Manager C++ API Database Connection Pool work?
The Database Connection Pool works similarly to the Datastore Pool, but the connection pool works transparently at the lower level of actual database connections to the Library Server database. This speeds the time for datastores to connect to the Library Server, and reduces the number of database connections to the absolute minimum of concurrently executing database operations, which is important to reduce the memory footprint and maximize scalability of the Library Server.

How do I configure and enable these features?
These features will be available in Content Manager v8.3 Fix Pack 2 or higher. For more detailed information, please read the readme file of the Content Manager release.

20.6 Additional resources
Performance tuning is a complicated, long term process. It will be involved in every stage of system's implementation, including planning, designing, and configuration. To perform performance tuning, the administrator needs extensive knowledge in these areas. In this chapter, we have introduced the related methods and steps.
For more details, refer to the following IBM Redbook and white papers.

- **Performance Tuning for Content Manager, SG24-6949**

  This redbook can be downloaded from the following Web site:

  This IBM Redbook deals with performance tuning for IBM DB2 Content Manager Version 8 for Multiplatforms. It is aimed at architects, designers, and system administrators of Content Manager systems.

  This book starts with an introduction to performance tuning basics. Then it defines what performance is and how it is measured, and describes performance methodology, the performance improvement process, along with a set of general guidelines you should use when planning a new system or maintaining and improving an existing system. In addition, it introduces a list of monitoring tools and performance tracing techniques that can be used to measure system performance and to help in discovering problems.

- **IBM DB2 Content Manager Enterprise Edition V8.3 Performance Troubleshooting Guide**

  This white paper can be downloaded from the following Web site:

  This document helps Content Manager system administrators troubleshoot performance issues for Content Manager V8.3 Enterprise Edition for DB2 and Oracle databases on the AIX, Solaris, and Windows operating systems. By using this guide, Content Manager administrators can use an organized and disciplined process that stays focused on identifying and resolving the primary system bottlenecks.

- **IBM DB2 Content Manager V8.3 Enterprise Edition Performance Monitoring and Maintenance Guide**

  This white paper can be downloaded from the following Web site:

  This document helps Content Manager Enterprise Edition V8.3 system administrators and IBM field support specialists monitor and maintain performance on their production systems by providing the tools and techniques recommended for monitoring the performance of a Content Manager system.

  By proactively monitoring a Content Manager system, administrators can identify and resolve most performance related issues before users are impacted, and identify system resource utilization trends for improved capacity planning. This document addresses Content Manager V8.3 Enterprise Edition V8.3 for DB2 and Oracle databases on the AIX, Solaris, and Windows operating systems.
- **IBM DB2 Content Manager V8.3 Enterprise Edition Performance Tuning Guide**

This white paper can be downloaded from the following Web site:

http://www.ibm.com/support/docview.wss?uid=swg27006452

This document presents best practices, tuning tips, techniques, and key tuning parameters to help you maximize performance for Content Manager Enterprise Edition Version 8.3 servers for DB2 and Oracle on the AIX, Sun Solaris, and Microsoft Windows operating systems. The topics covered include:

- An introduction of Content Manager Enterprise Edition Version 8.3 architecture.
- Performance tuning best practices and recommended performance methodology.
- Detailed performance tuning parameters and values for the DB2 and Oracle database servers, the operating systems, and techniques for monitoring performance for a Content Manager Version 8.3 system.
- Additional references to other performance tuning, troubleshooting, and monitoring resources.

- **Monitoring Content Manager V8.3 Enterprise Edition with DB2 Performance Expert V2.2 for MP**

This white paper can be downloaded from the following Web site:


This document will help Content Management system administrators use IBM DB2 Performance Expert for MP V2.2 to monitor and analyze the performance of Content Manager DB2 databases. This document provides guidelines for using Performance Expert in different Content Manager performance monitoring and analyzing tasks (for example, determining the reason for bad response times or discovering peak and normal workload times).
Troubleshooting

In this chapter, we offer advice on where to get Content Manager diagnostic information, which can be helpful when resolving problems. This includes how to set up traces that may be required by IBM support, where to find system information, and what you can do to get the system up again. There is also an explanation of the content of the traces. In addition, we discuss some common problems, and what you can do to avoid them.

The intent of this chapter is to provide a focused discussion on what actions should be performed when problems occur. You first learn how to pinpoint the problem to a specific area in the Content Manager system. Once the problem is pinpointed, we show you how to trace and resolve it.
21.1 Log and trace

For troubleshooting, one of the important tools is using log and trace files. The log and trace files help you to know the status, and to understand what happened.

In Content Manager V8.3, there are some improvements that are related to log and trace files. These improvements make it easier to troubleshoot and administer the product.

These improvements include:
- Single logging directory to make finding logs easier.
- Consolidated configurations to make setting up logging easier.
- Configure log through System Administration Client.
- Single-user only tracing.
- Consolidated log files and formats, especially in the eClient, beans, and APIs. Addition of a correlater ID to allow control flow to be followed across components and machines.

21.1.1 Single logging directory

In Content Manager V8.3, the common log location is the log directory in the working directory. The Content Manager working directory is located in the following directory:

- Windows: %IBMCMROOT% (default)
- UNIX: Home directory of the new system user ibmcmadm. For example: /home/ibmcmadm/

Table 21-1 compares the locations of log and trace files between Content Manager V8.2 and V8.3.

<table>
<thead>
<tr>
<th>Component</th>
<th>V8.2</th>
<th>V8.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library Server</td>
<td>/tmp/icmserver.log or c:\icmserver.log</td>
<td>&lt;working directory&gt;\log\ls&lt;dbname&gt;</td>
</tr>
<tr>
<td>Resource Manager and Services</td>
<td>WebSphere logs directory</td>
<td>&lt;working directory&gt;\log\rm&lt;node&gt;&lt;app&gt;</td>
</tr>
<tr>
<td>Connectors</td>
<td>Dklog.log</td>
<td>&lt;working directory&gt;\log\connectors\</td>
</tr>
</tbody>
</table>
### 21.1.2 Consolidated configuration files related to log settings

In Content Manager V8.3, the common configuration file location is the `cmgmt` directory in the working directory. The Content Manager working directory is located in the following directory:

- **Windows**: `%IBMCMROOT%` (default)
- **UNIX**: Home directory of the new system user ibmcmadm. For example: `/home/ibmcmadm/`

Table 21-2 compares the locations of configuration files between Content Manager V8.2 and V8.3.

<table>
<thead>
<tr>
<th>Component</th>
<th>V8.2</th>
<th>V8.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beans</td>
<td>eClientTrace.log</td>
<td><code>&lt;working directory&gt;\log\beans\</code></td>
</tr>
<tr>
<td>eClient</td>
<td>eClientTrace.log</td>
<td><code>&lt;working directory&gt;\log\eclient\</code></td>
</tr>
<tr>
<td>Windows Client</td>
<td><code>%APPDATA%\IBM\Content Manager\LOG\ICMClient.err</code> and <code>ICMClient.log</code></td>
<td><code>&lt;working directory&gt;\log\icmclient\&lt;username&gt;\</code></td>
</tr>
</tbody>
</table>

For more detailed information, refer to Appendix F, “Configuration and log files” on page 683.
For more detailed information, refer to Appendix F, “Configuration and log files” on page 683.

21.1.3 Configure log through System Administration Client

In Content Manager V8.3, you can configure most of log files through one GUI interface. This function is included in Content Manager System Administration Client.

For Windows platform, the steps to open the log configuration GUI window as below:

1. Select Start → Programs → IBM DB2 Content Manager Enterprise Edition → System Administration Client.
2. Select from the menu, Tools → Log Configuration.

After the two steps as above, you see the Log Configuration Utility GUI window as Figure 21-1.

![Log Configuration Utility GUI window](image)
21.1.4 Single-user only tracing

From the configuration log user interface (see Figure 21-1), you can enable single-user only tracing. The feature of single user only tracing has the following advantages:

- Enable troubleshooting for a single user.
- All other user IDs are not traced.
- All other user IDs get default ERROR logging.
- Minimizes performance impact of tracing.
- Enable full tracing in the Connectors, Library Server, Resource Manager for a given Content Manager User ID.

Note that the single-user only tracing takes effect on the next logon for the Content Manager user ID.

For more detailed information, refer to the Content Manager Information Center.

21.1.5 Consolidated log files and formats

In the Content Manager V8.3, there is a unique log correlater ID to log files that are related to the following components:

- Connector
- Library Server
- Resource Manager

The unique correlater ID is logged with each transaction to the Library Server, and map to related log files. The unique correlater helps an administrator to understand the status better.

The common content of log files include the following messages:

- GMT Timestamps
- Content Manager user name logged (Connector, Library Server and Resource Manager)
- Unique log correlater ID (Connector, Library Server and Resource Manager)
- Log messages

For more detailed information, refer to the Content Manager Information Center.
21.2 Pinpointing the problem

Understanding the overall architecture of the Content Manager system is extremely beneficial when troubleshooting errors. One of the first questions you should ask yourself when resolving a problem is, “Where in the system is this problem occurring, and which components are involved?” A DB2 Content Manager system involves many software components, some of which include the following software:

- DB2 UDB
- DB2 Net Search Extender (NSE)
- WebSphere Application Server
- IBM HTTP Server
- Content Manager Library Server
- Content Manager Resource Manager
- Windows Client

Almost all of these components are related in some way, and either directly or indirectly communicate with each other. For example, when performing a text search from the Windows client, the Content Manager Library Server, DB2 UDB, and DB2 NSE components are all used. If the text search fails, you must first determine in which component the failure occurred. This can usually be done by inspecting the error message being returned. In other cases, tracing must be performed.

The Library Server contains attributes (meta data), text search indexes, document routing information, and access control information. When a client performs a search, the Resource Manager is not involved. In fact, clients can logon and perform searches (whether they be parametric or text searches) even though the Resource Manager is not running.

If you are troubleshooting a problem related to searches or access control, the Library Server log file (see 21.2.1, “Library Server” on page 565 for information on the Library Server logs) is most likely the first place to begin your investigation. In some situations, you may see error messages containing DB2 return codes. In these circumstances, for an explanation of why this error is occurring, you should refer to:

- *IBM DB2 Universal Database - Message Reference Volume 1*, GC09-4840
- *IBM DB2 Universal Database - Message Reference Volume 2*, GC09-4841
The Resource Manager contains information regarding the storage of objects. As a Web application, the way at which you approach a Resource Manager problem is different from how you approach a Library Server problem. Due to the fact that the Resource Manager requires a valid token before it honors a request, ensuring that the Library Server has generated a valid token for the client is important (see 8.4, “Access to objects” on page 218 for more information on security tokens).

The log files from IBM HTTP Server and WebSphere Application Server must also be used when troubleshooting the Resource Manager. For example, a problem with the IBM HTTP Server configuration will cause client requests to the Resource Manager to fail (assuming the client is not bypassing the HTTP server, and it is directly accessing the Resource Manager).

The clients contain their own log files and tracing mechanisms for troubleshooting. When errors do arise, however, you need to determine if the problem is originated from the server or from the client. For example, if you get an error message when trying to import a document, you should check both the Resource Manager server and client log files.

In the remainder of this chapter, we show you how to troubleshoot for problems in the Library Server, Resource Manager, and clients. Content Manager Version 8 involves many components, and contains a fully functioned tracing facility. When a problem arises, take a moment to first determine where you should begin your investigation, doing so usually leads directly to the cause of the error.

### 21.2.1 Library Server

In this section, we show you how to troubleshoot Library Server related problems. The Library Server is a DB2 UDB database that is accessed by DB2 stored procedures. If there is an underlying problem with the database manager, problems with the Library Server will occur.

The Library Server configuration (see Figure 21-2) allows you to control the level and location of Library Server tracing. Use the Content Manager System Administration Client to update the Library Server configuration.
From the Library Server configuration window (see Figure 21-2), you can choose various trace levels. Specifically, this window allows you to choose between the following trace levels: basic, detailed, data, and performance. The level that you select specifies the maximum level of tracing that a client can request, as the administrator only enables tracing in this way; tracing is only done when requested by client applications.

When you change the trace level from the Library Server configuration window, the TRACELEVEL value in the ICMSTSYSCONTROL table is updated to a corresponding number. The valid trace level values are shown in Table 21-3.

**Table 21-3  Library Server trace levels**

<table>
<thead>
<tr>
<th>TRACELEVEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No trace</td>
</tr>
<tr>
<td>1</td>
<td>Basic trace: program flow is logged</td>
</tr>
<tr>
<td>2</td>
<td>Detail trace: both program flow and data are logged</td>
</tr>
</tbody>
</table>
As shown in Table 21-3, the TRACELEVEL options shown in the Library Server configuration window do not represent all possible options. Higher levels of tracing are set by updating the TRACELEVEL value in the Library Server database table named ICMTSTSYSCONTROL.

<table>
<thead>
<tr>
<th>TRACELEVEL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Data trace</td>
</tr>
<tr>
<td>8</td>
<td>Performance trace</td>
</tr>
<tr>
<td>15</td>
<td>All of the above</td>
</tr>
<tr>
<td>16</td>
<td>Build/parse</td>
</tr>
<tr>
<td>31</td>
<td>All of the above</td>
</tr>
<tr>
<td>32</td>
<td>Memory management</td>
</tr>
<tr>
<td>63</td>
<td>All of the above</td>
</tr>
<tr>
<td>256</td>
<td>Cache trace</td>
</tr>
<tr>
<td>512</td>
<td>Cache allocation</td>
</tr>
<tr>
<td>1024</td>
<td>Cache management</td>
</tr>
</tbody>
</table>

Note: A positive TRACELEVEL value enables tracing and sets the maximum tracing level that is available to clients. When a positive number is set, tracing is not enabled for the Content Manager System Administration Client. To force Content Manager to trace the System Administration Client actions (and all connections), you must set the tracing level to a negative value (such as -15), which can only be performed by manually updating the database table (see below).

As stated above, setting the TRACELEVEL to a positive number sets the maximum level of tracing (as can be set through the Content Manager System Administration Client for example) available to clients. A positive number enables the Content Manager Windows client actions to be traced, and the output is stored in the ICMSERVER.LOG file. If you have a custom client, you need to request tracing to be enabled for your client, merely setting the trace level to a positive number does not enable tracing for your client.

To determine the level of tracing to which the Library Server is currently set, you can issue the following commands from a DB2 command window, replacing the database name, user ID, and password with your values, if different from ours:

```
db2 connect to icmnlssdb user icmadmin using icmadmin
db2 select tracelevel from icmstsyscontrol
```
Example 21-1 shows a sample output that you should expect.

**Example 21-1  Viewing the Library Server trace level using DB2 Command Window**

```
C:\>db2 connect to icmnlsdb user icmadmin using icmadmin
   Database Connection Information
   Database server        = DB2/NT 8.1.0
   SQL authorization ID   = ICMADMIN
   Local database alias   = ICMMNLSDB
C:\>db2 select tracelevel from icmstsyscontrol
      TRACELEVEL
      ------------
          -15
          1 record(s) selected.
C:\>
```

To set the Library Server trace level from a DB2 Command Window use the following commands (again, replacing with your database name, user ID and password if necessary):

```
db2 connect to icmnlsdb user icmadmin using icmadmin
   db2 update icmstsyscontrol set tracelevel=<tracelevel>
```

Where:

- `<tracelevel>` represents a value from Table 21-3 on page 566.

By default, the Library Server traces are placed in the
%IBMCMROOT%\log\ls\icmnlsdb\ICMSERVER.LOG file on Windows and
/home/ibmcmadm/log/ls/icmnlsdb/ on UNIX. The name and location of this file
can also be changed from the Library Server configuration window (see
Figure 21-2 on page 566). Depending on the trace level value, the
ICMSERVER.LOG file contains different levels of information. In all cases,
however, the structure of the log file remains the same.

Example 21-2 show a piece of the ICMSERVER.log, which is the log file of the
Content Manager Library Server.

**Example 21-2  Sample of ICMSERVER.log**

```
ICMPLSRC ICMRMCHECKOUT 01653 11/27 23:56:02.205 GMT ;23005902171937
f81Xfd9c0962e3X-7ff8
ICMADMIN Exit rc=0 reason=0 extrc=0 extreason=0
......
......
ICMPLSLK ICMTraceInitServer 01067 12/02 23:39:11.234 GMT ;02175640235401 75:107ec9e5e02:X7fe1 ICMADMIN
The server has been paused for backup, please try again later.
*plRC= 7751
```
ICMTraceServer - Invalid data type for line 1
Hex data follows: 49434d54726163655365727665722076616c7565
ICMTraceServer - Invalid data type for line 2
Hex data follows: 49434d54726163655365727665722076616c7565ICMPLSLK
ICMTraceServer - Invalid data type for line 3
Hex data follows: 49434d54726163655365727665722076616c7565
ICMTraceServer - Invalid data type for line 4
Hex data follows: 49434d54726163655365727665722076616c7565
ICMTraceServer - Invalid data type for line 5
Hex data follows: 49434d54726163655365727665722076616c7565
ICMTraceServer - Invalid data type for line 6
Hex data follows: 000000000000000080ff80ff0000000000000000ICMPLSLK
ICMTraceInitServer 00463 12/02 23:40:17.671 GMT ;02175640235401
75:107ec9e5e02:X7fe1 ICMADMIN WARNING -- Previous SP (ICMPLSLK) ended without
proper exit, Verify SP name listed in icmplscm.ccc
ICMPLSDR ICMTraceInitServer 00463 12/02 23:43:00.937 GMT
;02001757597402 aeX107e8d515d7X-7dcf ICMADMIN WARNING -- Previous SP (ICMPLSDR)
ended without proper exit, Verify SP name listed in icmplscm.ccc

Each log entry consists of a stored procedure module name, DB2 stored
procedure name, program line number, date and time the log entry was made,
the process ID (useful when cross referencing DB2 logs), the unique log
correlater ID (if exist), and the description.

From the first record of Example 21-2 , we get the following information:

- ICMPPLSRC is the stored procedure module name
- ICMRMCHECKOUT is the DB2 stored procedure name
- 01653 is the program line
- 07/07 23:56:02.205 GMT is datetime
- 23005902171937 is the process id
- f81Xfd9c0962e3X-7ff8 is the correlater ID. We can use this correlater ID to
  find corresponding records from other related log files.

The point at which a stored procedure is called is logged with an Entry message.
Likewise, the point at which the stored procedure exits is logged with an Exit
message. Also note that the return code (rc) is shown when the stored procedure
exits. The stored procedure has exited successfully when the return code is
equal to zero (rc=0).

The Library Server log file is an important asset in resolving Library Server
issues. Reasons for not being able to create an item type or unacceptable query
performance can be determined by inspecting the Library Server log file. The
maximum trace level is not always needed to investigate a problem. Instead, you should use the minimum trace level that gives you enough information to determine the cause of the problem. Doing so keeps you from having to parse through extraneous and unrelated log information.

21.2.2 Resource Manager

In this section, we show you how to troubleshoot Resource Manager related problems. Configuration errors with WebSphere Application Server and/or DB2 UDB will cause the Resource Manager to fail. Being able to locate the failing component, and knowing how to correct the problem, is vital in maintaining a Resource Manager server.

When troubleshooting Resource Manager related problems, you should keep its architecture in mind. Understanding that the Resource Manager is a Web application is vital in being able to quickly analyze and resolve problems. Being a Web application, various components are involved when handling requests. For example, when a client requests a document, the Library Server, Web server, and Resource Manager Web application are all involved. You need to use the log files generated by each of these components in order to troubleshoot problems.

When investigating Resource Manager related problems, it is often helpful to think about the various different components involved and the how they are used, and in which order.

A client must first obtain a token from the Library Server. (Once a token is obtained, it can be reused by the client until it expires.) The client passes its request and this token to the HTTP Server. The WebSphere Plug-in (running inside the HTTP Server) forwards the request to the Resource Manager Web application (running inside of WebSphere Application Server). The Resource Manager Web application first decrypts and validates the token. Lastly, depending on the type of request, information is either read from or stored into the Resource Manager database (for example, rmdb) and file system volume (for example, Drive_C).

Attention: When accessing the Resource Manager from the Content Manager System Administration Client (for example, if you want to add a new file system volume), SSL must be properly configured since HTTPS is used instead of HTTP. However, SSL is not required when using the Windows client to retrieve and store documents.

In the remainder of this section, we show you how to systematically troubleshoot the Resource Manager. The topics include the following tasks:
Verifying database creation
Verifying database connections
Verifying Resource Manager deployment
Verify communication with Web server
Resource Manager logging
Secured Sockets Layer (SSL)

Verify database creation
If a new installation has been performed, and the Resource Manager is not functioning properly, you should ensure that the Resource Manager database was created successfully. The Content Manager installation log file is named cminstall.log and is located, by default, in %IBMCMROOT\log on Windows and /home/ibmcmadm/log on UNIX.

Review this file closely to be sure that all SQL commands were completed successfully. You must distinguish between error and warning messages, as both are contained in this log file. A common mistake is to forget to grant the necessary database authority to the Resource Manager user ID. This would result in the following message to be logged: *RMADMIN does not have the privilege to perform operation.*

After correcting any problems with your environment, it is possible to recreate the Resource Manager database, this avoids the need to start the installation for Content Manager again from scratch (the Library Server database may have been created without problems).

To recreate the Resource Manager database on Windows:
1. Select Start → Programs → IBM DB2 Content Manager Enterprise Edition → Resource Manager Database Install.
2. Follow the instructions provided by the utility program. Remember to take notes and write down the key names, user IDs, and passwords that you enter during this program.

To recreate the Resource Manager database on AIX:
1. Navigate to the DB2 Content Manager directory, for example:
   /opt/IBM/db2cmv8/config
2. Enter the command:
   `cmcfgrmdb`
3. Follow the instructions provided by the utility program. Remember to take notes and write down the key names, user IDs, and passwords that you enter during this program.
To recreate the Resource Manager database on Solaris:
1. Navigate to the DB2 Content Manager directory, for example:
   /opt/IBMicm/Config
2. Enter the command:
   cmcfgrmdb
3. Follow the instructions provided by the utility program. Remember to take notes and write down the key names, user IDs, and passwords that you enter during this program.

Verify Resource Manager deployment
Being able to verify that the Resource Manager Web application has been successfully deployed is vital in resolving Resource Manager installation problems. If the Web application was not installed properly, or if the Web server plug-in was not regenerated, then the Resource Manager server will be unresponsive.

As Figure 21-3 shows, a client can access the Resource Manager Web application (icmrm) in two ways. The first is the default method of going through the Web server (for example, IBM HTTP Server). In this case, a request to port 80 is made by http://server/icmrm/snoop. The Web server plug-in forwards the request to WebSphere Application Server. The second method is to send the request directly to WebSphere Application Server by specifying the port the application server instance is listening on (for example, http://server:9080/icmrm/snoop).

![Diagram of Web server integration with WebSphere Application Server]

Figure 21-3 Web server integration with WebSphere Application Server
If the direct method (http://server:9080/icmrn/snoop) fails, then either the Resource Manager Web application is not started, or has not been properly deployed. In this circumstance, manually deploying the Resource Manager Web application may resolve the problem. For instructions on manually deploying the Resource Manager application, see IBM DB2 Content Manager for Multiplatforms - Revised Installation Steps for Windows. There is a revised installation guide for both AIX and Solaris too.

If the direct method is successful, but going through the Web server (http://server/icmrn/snoop) fails, then the problem lies with the Web server plug-in. In this circumstance, regenerating the Web server plug-in (using the WebSphere Application Server Administrative Console — server1 needs to be running) usually resolves the problem (see Figure 21-4).

**Important:** When you have regenerated the Web server plug-in, restart your Web server so that it can pick up the new plug-in configuration.
To regenerate the Web server plug-in:

1. Log into your WebSphere Application Server Administrative Console. By default, this is at http://<hostname>:9090/admin.

2. Expand Environment in the left hand panel.


4. Select OK in the right hand panel.

5. Restart your Web server.

**Note:** The WebSphere standard error log file may also contain error messages related to the Resource Manager deployment and operation. The standard error log file is located by default, in:

- C:\Program Files\WebSphere\AppServer\logs for Windows
- /usr/WebSphere/AppServer/logs for AIX
- /opt/WebSphere/AppServer/logs for Solaris

### 21.2.3 Problems starting WebSphere Application Server on AIX 5L

There can be problems starting WebSphere Application Server on AIX 5L™. We have encountered problems starting server1 on AIX 5.1 and 5.2, due to a port conflict. When we look at the SystemOut.log for server1, located in /usr/WebSphere/AppServer/logs/server1, we find the following error:

```
[9/11/03 12:57:21:195 PDT] 51a3cd57 WebContainer  E SRVE0146E: Failed to Start Transport on host , port 9090. The most likely cause is that the port is already in use. Please ensure that no other applications are using this port and restart the server.
```

This is due to an AIX administration tool using port 9090, which is a default transport port for WebSphere Application Server. When we look in our /etc/services file, we find a service known as wsmserver currently using port 9090 (see Figure 21-5).
Figure 21-5  AIX port conflict with WebSphere Application Server

To get around this problem, you can either change the port that the AIX Web-based Systems Manager Server (WSM) uses; or, if you do not want to change your current AIX environment, you can change the transport port that WebSphere Application Server uses from 9090 to a different port, 9091 for example. To do this use the following steps:

1. Identify a port that is not used currently, check in the /etc/services file to see if the port you have in mind is already used, we chose 9091.

2. Edit the file server.xml as root, which is located in /usr/WebSphere/AppServer/config/cells/<hostname>/nodes/<hostname>/servers/server1 and locate the following section (execute a find on 9090 to locate this section quickly):

   ```xml
   <transports xmi:type="applicationserver.webcontainer:HTTPTransport" xmi:id="HTTPTransport_3" sslEnabled="false">
     <address xmi:id="EndPoint_3" host="" port="9090"/>
   </transports>
   ```

3. Update the port setting to the number of the port you wish to change it to, for example, `port="9091"`.

4. Save the file.

5. Edit the file virtualhosts.xml as root, which is located in /usr/WebSphere/AppServer/config/cells/<hostname>/servers/server1 and locate the following section (it is right at the bottom of the file):

   ```xml
   <aliases xmi:id="HostAlias_4" hostname="*" port="9090"/>
   ```

6. Update the port setting to the same number port that you specified in the server.xml file.

7. Save the file.

8. Start server1 - the server should now start.
We edited the XML files directly in the foregoing example, as we were unable to use the WebSphere Administrative Console due to the fact that server1 cannot be started.

**Verify database connections**
When the Resource Manager Web application starts, it attempts to connect to the Resource Manager database (rmdb). If this database connection cannot be made, the Resource Manager will be unable to handle client requests.

**Note:** When the Resource Manager starts, it makes three connections to the Resource Manager database (rmdb by default).

To validate that the database connections are active, from a DB2 Command Window run:

```
db2 list applications
```

Three connections to the Resource Manager database should appear. (See Figure 21-6: Only connections with an Auth Id of RMADMIN are related to the Resource Manager.)

![Figure 21-6: Resource Manager database connections at start up time](image)

If the connections do not appear, then the Resource Manager Web application is having a problem connecting to the database. Usually, this occurs when the user ID and password used by the Resource Manager to connect to the database are invalid. The user ID and password information are stored in the ICMRM.properties file located in:
On Windows:
C:\ProgramFiles\WebSphere\AppServer\installedApps\\<hostname>\icrmrme.ear\icrmr.ear\WEB-INF\classes\com\ibm\mm\icrmr

On AIX: /usr/WebSphere/AppServer/...

On Solaris: /opt/WebSphere/AppServer/...

Validate the values for the DBUserid and DBPassword parameters. (Enter the password as plain text, and the Resource Manager will re-encrypt it.) The user ID and password entered here can be tested by issuing the following command from a DB2 Command Window:

db2 connect to rmdb user <userid> using <password>.

Database connections will also fail if the db2java.zip file is not in the WebSphere Application Server classpath. In this circumstance, the WebSphere Application Server standard error log file will contain a message indicating that the DB2 JDBC driver could not be found.

Verify communication with Web server

If the Resource Manager hostname was incorrectly specified during the installation (or if the Resource Manager hostname has changed), a client request will never reach the Web server. By default, the IBM HTTP server logs every client request in:

C:\IBM HTTP Server\logs\access.log file for Windows

C:\usr\IBMHttpServer/logs/access_log for AIX

C:\opt\IBMHTTPD/logs/access_log for Solaris

This file can be used to verify the Resource Manager URL and that the client request is getting to the Web server. Example 21-3 shows sample content for the access.log file when a client imports a document.

Example 21-3  Example content for IBM Http Server access.log file

| 103.166.1.99 - - [17/ Sep/ 2003:21:07:06 -0500] | "POST /icrmr/ ICMResourceManager HTTP/1.1" 200 403 |

The URL used by the client to access the Resource Manager can be configured from the Content Manager System Administration Client. Choosing a Resource Manager and selecting its properties will display the window shown in Figure 21-7. From this window, you can specify the hostname, port, and URL path.
21.2.4 WebSphere global security

If your Resource Manager application server does not restart after you enable global security, you can disable security:

1. Go to your $<install_root>\bin directory.
2. Execute the following command:
   
   wsadmin -conntype NONE

3. At the wsadmin> prompt, enter securityoff and then type exit to return to a command prompt.
4. Restart the server with security disabled to check any incorrect settings through the administrative console.

Web applications that use J2EE FormLogin style login pages (such as the WebSphere Application Server administrative console) require single sign-on (SSO) enablement. You should only enable SSO (see Figure 21-8) for certain advanced configurations where LTPA SSO type cookies are not required. For this
reason, if you disable SSO, you will not be able to logon to the WebSphere Application Server Administrative Console, and will have to follow the steps in the paragraph above in order to gain access to the console.

![WebSphere Administrative Console - Microsoft Internet Explorer](image1.png)

**Figure 21-8 Single sign-on enabled when using global security in certain situations**

### 21.2.5 Resource Manager logging

The Resource Manager logs errors into the log file named icmrm.logfile. This file is located, by default, in:

- `<IBMCMROOT>/log/rm/<WASNode>/<WASApp>/`
  - `<IBMCMROOT>` is the directory where the Content Manager is installed.
  - `<WASNode>` is the node name of Websphere Application Server.
  - `<WASApp>` is the application name of the Resource Manager.

In addition to the default log file, the Resource Manager contains a logging facility that is based upon Log4J (an open source project available from apache.org). The logging facility consists of various xml FILES, each of which controls the level and location of logging for different parts of the Resource Manager server. A description of these log files can be found in Table 21-4.
Table 21-4 Resource Manager logging control files

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icmrm_asyncr_logging.xml</td>
<td>Logging control for asynchronous recovery utility</td>
</tr>
<tr>
<td>icmrm_logging.xml</td>
<td>Logging control for Resource Manager servlets</td>
</tr>
<tr>
<td>icmrm_migrator_logging.xml</td>
<td>Logging control for migrator process</td>
</tr>
<tr>
<td>icmrm_purger_logging.xml</td>
<td>Logging control for purger process</td>
</tr>
<tr>
<td>icmrm_stager_logging.xml</td>
<td>Logging control for stager process</td>
</tr>
<tr>
<td>icmrm_replicator_logging.xml</td>
<td>Logging control for replicator process</td>
</tr>
<tr>
<td>icmrm_validator_logging.xml</td>
<td>Logging control for validator process</td>
</tr>
</tbody>
</table>

By default, the Resource Manager logging control files are located in:

- `<IBMCMROOT>/cmgmt/rm/<WASNode>/<WASApp>/` on Windows
  - `<IBMCMROOT>` is the directory where the Content Manager is installed.
  - `<WASNode>` is the node name of Websphere Application Server.
  - `<WASApp>` is the application name of the Resource Manager.

Problems can be traced by adjusting the logging level in the respective XML file.

To trace a problem, open a logging control file and locate the following two lines (located towards the end of the file):

```
<priority value="INFO" class="com.ibm.mm.icmrm.util.ICMRMPriority"/>
<appender-ref ref="ASYNC"/>
```

These are the only two lines that you need to change. The priority parameter specifies the level of tracing. The valid priority values are described in Table 21-5.

Table 21-5 Resource Manager logging priority values

<table>
<thead>
<tr>
<th>Priority value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FATAL</td>
<td>Only log if the servlet is terminating unexpectedly.</td>
</tr>
<tr>
<td>ACTION</td>
<td>Messages that describe an action the system administrator needs to take. These are not errors, but conditions such as being short on space.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Indicates a request was unable to be fulfilled or an internal error.</td>
</tr>
<tr>
<td>WARN</td>
<td>Warnings of unexpected behavior.</td>
</tr>
</tbody>
</table>
For example, if you wanted to log all possible information, you can update the priority tag to look like:

```xml
<priority value="DEBUG" class="com.ibm.mm.icmrm.util.ICMRMPriority"/>
```

The appender name parameter specifies where log messages are written. The valid appender name values are described in Table 21-6.

**Table 21-6   Resource Manager logging appender-ref values**

<table>
<thead>
<tr>
<th>Appender-ref value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>Messages are sent to a log file. The filename is specified in the FILE stanza (located toward beginning of xml file). By default, the file will be placed in the C:\Program Files\WebSphere\AppServer\logs directory for Windows, /usr/WebSphere/AppServer/logs for AIX and /opt/WebSphere/AppServer/logs for Solaris.</td>
</tr>
<tr>
<td>WRAP</td>
<td>Messages are sent to a circular log file (after the file reaches a maximum size, the oldest messages are removed to make room for the most recent messages). By default, the file will be placed in the C:\Program Files\WebSphere\AppServer\logs directory for Windows, /usr/WebSphere/AppServer/logs for AIX and /opt/WebSphere/AppServer/logs for Solaris.</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>Messages are sent to standard output, which ends up in the WebSphere log files.</td>
</tr>
</tbody>
</table>

**Note:** Increasing the logging priority value may impact the performance of the Resource Manager.
For example, to send messages to a log file, the appender name tag should look similar to either one of the following tags:

```xml
<appender name="WRAP" class="org.apache.log4j.RollingFileAppender">
<appender name="FILE" class="org.apache.log4j.FileAppender">
```

The log filename will consist of icrmr.<component>.logfile, where `<component>` is the Resource Manager process that is writing to the log file.

For example, icrmr_migrator_logging.xml will create a file named icrmr.migrator.logfile.

In addition to the logging facility, the Resource Manager provides an administrative servlet. To access this servlet:

1. Open a Web browser.
2. Go to https://<hostname>/icrmr/ICMRMAdminServlet, where `<hostname>` is the name of your machine (see Figure 21-9).
   
   SSL connections do not work with the localhost hostname.
3. Login as rmadmin, using password as the password.

From this browser interface, you can update the Resource Manager configuration and view the object storage configuration.

<table>
<thead>
<tr>
<th>Appender-ref value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASYNC</td>
<td>Messages are sent to standard output, which ends up in the WebSphere log files. While this is faster than CONSOLE, it does not include file and line number information in its messages.</td>
</tr>
</tbody>
</table>
21.2.6 Secured Sockets Layer (SSL)

A secured sockets layer (SSL) is only required to perform Resource Manager configuration. If you are having problems importing or retrieving documents, you can safely conclude that SSL is not the cause. If you are trying to access the Resource Manager from the System Administration Client, and are receiving an error, then SSL may be the culprit.

Steps to configure SSL can be found in *IBM DB2 Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332. When configuring SSL, you must create a self-signed certificate and configure the Web server for use with SSL. Also, if you are using WebSphere Application Server V4 AE, you must add *.443 as a virtual host alias (configured via the WebSphere Administrative Console, see 21.2.7, “Generating the Web server plug-in with SSL information for WebSphere Application Server” on page 584 for full instructions).
When using WebSphere Application Server V5, make sure that port 443 is specified within a host alias for your machine name, within the virtual hosts settings for the default host, as shown in Figure 21-10 (see 21.2.7, “Generating the Web server plug-in with SSL information for WebSphere Application Server” on page 584 for full instructions). If it is not, then you need to add this definition, in order for the Content Manager System Administration Client to Resource Manager communication to work.

![Figure 21-10 Adding a host alias to the default host in WebSphere v5](image)

### 21.2.7 Generating the Web server plug-in with SSL information for WebSphere Application Server

Use the following steps to generate the Web server plugin with SSL information for WebSphere Application Server.
**WebSphere Application Server V5**

The following steps are used to generate Web server plug-in with SSL for WebSphere Application Server V5:

1. In a Web browser, launch the following URL:
   
   http://your_hostname:9090/admin

2. Type wasadmin (or your particular user ID and password if you have WebSphere global security enabled) and select OK. The WebSphere Application Administrative Console opens.

3. In the left frame, expand **Environment** and select **Virtual Hosts**.

4. In the right frame, select **default_host**, then select **Host Aliases** and select **New**.

5. In the Host Name field, type * in the Port field, type 443. Select OK. Select **Save**.

6. Select the **Save** button (to save all changes).

7. In the left frame, select **Update Web Server Plugin**. In the right frame, select **OK**.

8. Select **Logout** and close the Web browser.

9. Restart the IBM Http Server as follows:
   
   - On Windows:
     
     Stop and restart the HTTP Server service
   
   - On AIX:
     
     /usr/IBMHttpServer/bin/apachectl graceful
   
   - On Solaris:
     
     /opt/IBMHttpServer/bin/apachectl graceful

10. Stop and restart WebSphere Application Server as follows:

    - On Windows:
      
      Stop and restart the WebSphere Application Server service

    - On AIX:
      
      /usr/WebSphere/appServer/bin/stopServer.sh server1
      /usr/WebSphere/AppServer/bin/startServer.sh server1

    - On Solaris:
      
      /opt/WebSphere/AppServer/bin/stopServer.sh server1
      /opt/WebSphere/AppServer/bin/startServer.sh server1
WebSphere Application Server V4 Advanced Edition (AE)
The following steps are used to generate Web server plug-in with SSL for WebSphere Application Server V4:

1. Make sure that the WebSphere Application Server (WAS) service is started.
2. Invoke the WebSphere Application Administrative Console.
3. Select **Virtual Hosts** in the tree on the left frame of the console, then select the **General** tab on the right frame of the console Click **Add**.
4. Enter `*:443` in the text area that appears (that is an asterisk, a colon, then the numbers 443).
5. Select **Apply**.
6. Select **Nodes** (to expand that part of the tree).
7. Right-click `<your hostname>` in the tree on the left frame.
8. Select **Regen Webserver Plugin**.
9. Restart the IBM HTTP Server and the WebSphere Application Server so that the latest plugin information takes effect.

A Web browser can be used to test the SSL configuration at various points in the system. Before troubleshooting SSL, be sure that your Resource Manager is operating properly. This can be accomplished by either importing or retrieving a document though the Content Manager Windows client.

Once you have verified the Resource Manager configuration, open a Web browser and go to `https://<hostname>`, where `<hostname>` is the hostname of your Web server. Notice that `https` is used instead of `http`.

After accepting the self-signed certificate you created during the SSL configuration (see Figure 21-11), the IBM HTTP Server welcome page should appear. If you instead get an error message, check the IBM HTTP Server log file named `error.log` for SSL related error messages. This log file is located, by default, in `C:\IBM HTTP Server\logs`. Within this log file you should see a message that indicated why the SSL is not working. Typically this is a misspelled key file name or certificate name.
After verifying the SSL connection between the client and HTTP server, you should then validate the SSL connection between the client and WebSphere Application Server. In this case, you must specify the default SSL port of 443. This is accomplished by opening a Web browser and going to https://<hostname>:443/icmrm/snoop. If this fails, be sure to check the WebSphere Application Server log files which are located, by default, in:

- C:\Program Files\WebSphere\AppServer\logs for Windows
- /usr/WebSphere/AppServer/logs for AIX
- /opt/WebSphere/AppServer/logs for Solaris

Lastly, you should verify the SSL connection is working when communicating from the client, through the Web server, to WebSphere Application Server. This is accomplished by opening a web browser and going to https://<hostname>/icmrm/snoop. If this fails, you should check the IBM HTTP Server log files which are located, by default, in:

- C:\IBM HTTP Server\logs for Windows
- /usr/IBMHttpServer/logs for AIX
- /opt/IBMHTTPD/logs for Solaris

### 21.2.8 Clients

In this section, we show you how to troubleshoot problems relating to the Windows Clients. Log files and the procedure for performing tracing are discussed. Analyzing these traces and error logs help you determine where the error originated.
Windows client
The Content Manager Windows client is built with the C++ Object Oriented API Toolkit. In most circumstances, problems with the Windows Client can be attributed to problems connecting to the Library Server, or problems accessing the Resource Manager. The connection to the Library Server is made using the DB2 Runtime Client.

When a client logs on to a Library Server, two connections are made. The first is the physical database connection to ICMNLSDB (or whatever you named your Library Server database). The second is the logical connection to Content Manager (where the supplied user ID and password is authenticated with what is stored in the ICMSTUSERS table).

In order for the database connection to be made, the Library Server database must be cataloged on the client workstation (which is why the DB2 Runtime Client is needed). To check if the Library Server database has been cataloged on the client machine, you can go to a DB2 Command Window by (Start → Programs → IBM DB2 → Command Line Tools → Command Window) and enter: `db2 list database directory`.

The Library Server database should appear. If you do not see the database listed, you can use the DB2 Client Configuration Assistant (Start → Programs → IBM DB2 → Set-up Tools → Configuration Assistant) to catalog the database on the client machine.

Once the database is cataloged on the client machine, you should validate the connection by going to a DB2 Command Window and running `db2 connect to icmnlsdb user icmconct using password`. By default, icmconct is the database connection user ID to be used by clients. The user ID and password are stored, in encrypted format in the cmbicmenv.ini file. This file can be updated from the System Administrator Client, by selecting Tools → Manage Database Connection ID from the menu bar.

Any errors that occur while using the Windows Client will be logged in the log file directory. To determine what the log file directory is, select Options → Preferences from the menu bar, (see Figure 21-12).
Figure 21-12  Windows client log file location

The error log file is named ICMClient.err, and contains detailed error messages which are useful for troubleshooting. For example, when a document import fails, the following messages may be found in this file, as shown in Example 21-4.

Example 21-4  Sample error log file, ICMClient.err

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003-10-01 12:10:53.739</td>
<td>[2852] viitem : Exception DKXDOError (-1) in DKDDO::add()</td>
<td></td>
</tr>
<tr>
<td>2003-10-01 12:10:53.749</td>
<td>[2852] viitem : Error State:</td>
<td></td>
</tr>
<tr>
<td>2003-10-01 12:10:53.749</td>
<td>[2852] viitem : Error text: ICM9804: The security token supplied with order store was invalid.::HTTP/1.1 204 No Content (SERVER RC): 9804</td>
<td></td>
</tr>
<tr>
<td>2003-10-01 12:10:53.769</td>
<td>[2852] viitem : Exception Class Name: DKXDOError</td>
<td></td>
</tr>
</tbody>
</table>
The important text is in bold. An explanation and action plan for the ICM9804 message can be found in *IBM DB2 Content Manager for Multiplatforms - Messages and Codes*, SC27-1349. This particular error can be resolved by going to the Library Server configuration and choosing to regenerate the encryption key. (Be sure the Resource Manager servers are running when you regenerate the encryption key.) Refer to “Encryption key management” on page 221 for instructions on how to regenerate the encryption key.

In the Windows Client configuration file directory, you also find a file named ICMClientLog.ini. This file allows you to enable or disable tracing for different client components. For example, if you are experiencing problems with the login dialog, you can update the value for LOGINDLG from d to e. Trace messages are logged to the ICMClient.log file.

### 21.2.9 Installation

In Content Manager V8.3, several improvements are made to make it easier to install Content Manager.

1. **Prerequisites button through LaunchPad.**

   The Content Manager V8.3 installation LaunchPad has a Prerequisites button. This option helps you to validate whether the current system has the correct prerequisite software installed. When you click **Prerequisites**, you will be shown a list of the prerequisite software found on the current system. This listing will not tell you if the prerequisite software found on the system meets the minimum requirements or not. You should compare the listing of prerequisite software found on your system against the minimum requirements specified in the documentation.

2. **Elimination of the C++ compiler dependency.**

   In previous releases, the product configuration was very sensitive to the exact set up of the C++ compiler, and the product configuration would fail if the C++ compiler environment did not precisely match what was required. In Content Manager V8.3, the C++ compiler requirement is removed entirely. This makes the installation of Content Manager less complicated.

3. **User ID detection and creation.**

   Several operating system user IDs are required for the configuration of Content Manager. In Content Manager V8.3, the installation checks if the appropriate user IDs exist and it offers to create these user IDs if they are not present on the system.
– If the user ID exists, the installation use this existing user ID. Note that the installation program will not make changes to existing user IDs or to the UNIX .profile for existing users. If you want to use an existing user ID, ensure that it is set up correctly. For more detailed information, refer to Planning and Installing Your Content Management System, GC27-1332.

– If the user ID does not exist, the installation will ask if you want the user ID created for you. If you answer yes, then the installation program will create the user ID, add the user ID to the correct groups, and create the .profile as needed.

4. Automated SSL configuration option for Content Manager.

Previous releases of Content Manager required you to manually configure SSL for the Resource Manager. The manual configuration steps are tricky and error-prone. Content Manager V8.3 installation is improved to provide the option for the SSL configuration for you.

The functionality listed above highlights the installation improvements that will help you to smoothly install Content Manager.

For more detailed information, refer to the DB2 Content Manager manual Planning and Installing Your Content Management System.

Once you run the Content Manager installation, you can take advantage of several improvements for validating and troubleshooting your system’s installation.

1. Installation validation utility
2. Log for the Content Manager installation
3. Debug logging for the Content Manager installation

Let us go through each of these areas in more detail.

**Installation validation utility**

After Content Manager V8.3 has been installed, there is an option to launch the installation validation utility. You can also run the utility outside of the installation program later.

To run the installation validation utility outside of the installation program, run the following command:

```bash
<IBMCMMROOT>/bin/cminstvu
```

On Windows, you can also launch the utility through the Start menu.
This utility performs several basic checks of the components that were just installed, such as the following:

- Does the database exist and can it be connected to?
- Are key values in the database set?
- Is the Web application deployed and do HTTP and HTTPS connections work?
- Do the configuration files exist?

This utility also performs some verifications of system connectivity. These connectivity checks may fail if a distributed configuration does not have all machines set up yet.

**Log for the Content Manager installation**

For the installation of Content Manager, the locations of the log files are listed in Table 21-7.

<table>
<thead>
<tr>
<th>Log locations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Install Log</strong></td>
</tr>
<tr>
<td>CM</td>
</tr>
<tr>
<td>II4C</td>
</tr>
<tr>
<td>eClient</td>
</tr>
</tbody>
</table>

The log file cminstall.log records the processes of the Content Manager installation. Additionally, there are secondary log files in the IBMCMROOT/log/install directory.

The most recent installation log has the name shown in Table 21-7. When you reinstall a product on the system, the previous log file is backed up in the same directory with a new log file name. The new log file name is the original log file name plus a timestamp. The timestamp reflects the time when the log file was renamed.

For example:

cminstall.log_2005-12-10_13.10.58

**Debug logging for the Content Manager installation**

To turn on debug logging, follow the steps below:

1. Set the environment variable IBM_CM_DEBUG=<product>.
2. Run the installation program from the same command window

Where <product> is CM, IIC, EC, or FP. The FP abbreviation is for fix packs.
For example, to enable debug logging of the Content Manager installation on the Windows platform, follow these steps:

1. Open a DOS command prompt, and change directory to the location where the installation program is located.

2. Run the following two commands from the same command prompt:
   a. `set IBM_CM_DEBUG=CM`
   b. `install_CM`

For more detailed information, refer to Chapter 50 in *Planning and Installing Your Content Management System*.

### 21.3 Troubleshooting TSM integration

In this section, we introduce various resources for troubleshooting problems with a TSM integration. By utilizing log files, traces, and the TSM Backup Archive client, you can quickly and easily resolve almost any configuration error. The following resources is discussed:

- Resource Manager migrator log
- TSM Client API log
- TSM Backup Archive Client

#### Resource Manager migrator log

The Resource Manager server contains a logging facility based upon the open source Log4J project. A detailed description of this logging facility and the various logging options can be found in 21.2.5, “Resource Manager logging” on page 579. The logging control file for the migrator process is named `icmrmm_migrator_logging.xml`. This file is located, by default, in:

- `C:\Program Files\WebSphere\AppServer\installedApps\<hostname>\icmrmm.ear\icmrmm.war` on Windows
- `/usr/WebSphere/AppServer/installedApps/<hostname>/icmrmm.ear/icmrmm.war` on AIX
- `/opt/WebSphere/AppServer/installedApps/<hostname>/icmrmm.ear/icmrmm.war` on Solaris

To trace the migrator process, open `icmrmm_migrator_logging.xml` in Notepad (or equivalent), and locate the following two lines (towards the end of the file):

```xml
<priority value="INFO" class="com.ibm.mm.icmrmm.util.ICMRRMPriority"/>
<appender-ref ref="ASYNC"/>
```
To obtain detailed debugging information, change the priority value from INFO to DEBUG. To send log messages to a file, change the appender-ref value from ASYNC to FILE. Restart the Resource Manager Web application and migrator process. Detailed messages concerning the migration of documents are logged in a file named icmrm.migrator.logfile. This file is located, by default, in:

- C:\Program Files\WebSphere\AppServer\logs on Windows
- /usr/WebSphere/AppServer/logs on AIX
- /opt/WebSphere/AppServer/logs on Solaris

**TSM Client API log**

While the Resource Manager logging facility is useful in determining that a problem does exist, it may not always be the most helpful in telling you the exact details of what is causing the problem. For example, in certain situations, the Resource Manager logs may only contain information alluding to a connection problem with the TSM server. Details on what is causing the connection problem may not necessarily be in the Resource Manager log files.

The Resource Manager (and its processes) use the TSM Client APIs to communicate with the TSM server. By default, when an error occurs, the TSM Client APIs will log the error information in a file named dsierror.log. The information contained in this file proves invaluable in locating and resolving problems which deal solely with TSM, and have nothing to do with the Resource Manager configuration. Errors such as expired passwords and invalid management class names will be logged in this file.

**Note:** When troubleshooting a Tivoli Storage Manager integration problem, the dsierror.log file usually contains the cause of the error.

One common error which you may see in the dsierror.log file is that the dsm.opt file cannot be located. Recall that the TSM Client APIs use the DSMI_CONFIG variable to determine which option file to use. If there is a syntax error in the option file you specified, or if the file cannot be opened, then the TSM Client APIs will fallback to using the default option filename of dsm.opt. If you receive this type of message, you probably have a typo in your .opt file.

**TSM Backup Archive Client**

In some situations, the Resource Manager and TSM Client API log files may indicate that an invalid TSM server configuration is keeping documents from being migrated. One of the easiest ways to test a TSM server configuration is to use the TSM Backup Archive Client to back up a file.
The TSM Backup Archive Client also uses a set of variables to determine which client option file to use. More specifically, the DSM_CONFIG environment is used by the TSM Backup Archive Client to determine which client option file to use. This variable should point to the client option file used by the Resource Manager. Doing so will ensure the backup client is using the same TSM configuration as the Resource Manager.

If the TSM Backup Archive Client fails to back up a file, then the Resource Manager server will also fail. (Remember, the Resource Manager server does not archive files; it backs up files.) This indicates a problem with the TSM Server configuration. In this circumstance, you should check the TSM servers activity log to determine what the problem is.

**TSM Server**

If you are encountering errors still, there are some things that you should check on the Tivoli Storage Manager server using the administrative client.

Run the command `query node nodename f=d`, replacing nodename with the nodename that is being used to back up DB2 Content Manager objects. The values should be set as follows:

- **Backup deleted allowed:** This value should be set to YES.
- **Maximum Mount Points Allowed:** This value should be equal to or less than the number of drives on the Tivoli Storage Manager server.
- **Locked?:** This value should be set to NO.
- **Policy Domain Name:** This value should be checked to make sure it belongs to the correct domain. It determines which management classes are available to the node for managing the Content Manager object backups.

Using the value of Policy Domain Name, you can run some queries to see if the correct management classes for the Content Manager object backups are available. The command `query mgmtclass <domain> active` shows which is the DEFAULT management class for the node. The command `query copygroup <domain> active` tells you what the retention settings are for the management classes.

### 21.4 Traces on z/OS

With Content Manager for z/OS V8.3, the logs are no longer written to the CICS® CEEOUT, because there is no more CICS used. There are new locations for the logs.
21.4.1 Library Server log

The Library Server log is now written to the SYSPRINT of the WLM of the Library Server. To enable the trace, you must set the value in the ICMSTSysControl table of your Library Server database. Table 21-8 shows the values and the associated level of tracing.

Table 21-8  Library Server trace levels

<table>
<thead>
<tr>
<th>Decimal value</th>
<th>Hexadecimal value</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x000000</td>
<td>ICMTRACE_OFF</td>
</tr>
<tr>
<td>1</td>
<td>0x000001</td>
<td>ICMTRACE_BASIC</td>
</tr>
<tr>
<td>2</td>
<td>0x000002</td>
<td>ICMTRACE_DETAILED</td>
</tr>
<tr>
<td>4</td>
<td>0x000004</td>
<td>ICMTRACE_DATA</td>
</tr>
<tr>
<td>8</td>
<td>0x000008</td>
<td>ICMTRACE_PERFORMANCE</td>
</tr>
<tr>
<td>16</td>
<td>0x000010</td>
<td>ICMTRACE_PARSEBUILD</td>
</tr>
<tr>
<td>32</td>
<td>0x000020</td>
<td>ICMTRACE_MEMDEBUG</td>
</tr>
<tr>
<td>256</td>
<td>0x000100</td>
<td>ICMTRACE_CACHE_TRACE</td>
</tr>
<tr>
<td>512</td>
<td>0x000200</td>
<td>ICMTRACE_CACHE_ALLOC</td>
</tr>
<tr>
<td>1024</td>
<td>0x000300</td>
<td>ICMTRACE_CACHE_MGT</td>
</tr>
</tbody>
</table>

Note: A positive trace level value enables tracing and sets the maximum tracing level that is available to clients. When a positive number is set, tracing is not enabled for the Content Manager System Administration Client. To force Content Manager to trace the System Administration Client actions (and all connections), you must set the tracing level to a negative value, which can only be performed by manually updating the database table.

If you have a custom client, you need to request tracing to be enabled for your client, merely setting the trace level to a positive number does not enable tracing for your client.
21.4.2 Resource Manager trace

The Resource Manager trace information is written to the HTTP server's SYSPRINT dataset. Whether or not the trace information is written to this dataset and what information is written to it depends on two setups.

First, to get any output in the SYSPRINT at all, in your ICSPARM parameter of your HTTP server startup JCL, you must include the -v or -vv parameter, so that it looks similar to the following coding:

```
ICSPARM='-vv -p 8080 -r /etc/httpd.conf.8080'
```

Without this parameter, no messages will be written to the SYSPRINT.

Second, the TRACELEVEL setup in the ICMRMControl table. If this value is set to 0, only trace messages are written when an error occurs. A value of 1 is for debugging purposes and causes additional messages to be printed at the entry and exit of each Resource Manager subroutine.

**Note:** Do not forget to set the value back to 0 when you have solved the problem; otherwise, a large amount of data will be continuously written to your SYSPRINT dataset.

21.4.3 Workstation log

The logs on the workstation site are the same as described in 21.2.8, “Clients” on page 587. Refer this section for information.
Appendixes

In this part of the book, we present the following supplementary information:

- Appendix A, “Installation and configuration overview (for V8.2 only)” on page 601
- Appendix D, “ACL user exits UDF declarations” on page 665
- Appendix E, “API migration tables for Content Manager” on page 669
- Appendix F, “Configuration and log files” on page 683
Installation and configuration overview (for V8.2 only)

In this chapter, we describe the process of installing, customizing, and configuring a Content Manager V8.2 system. We cover the process for all the platforms, including Windows, AIX, Sun Solaris, and z/OS.

Important: The installation and configuration steps described in this chapter are specific for Content Manager V8.2 only. It is not to be used to install and configure the latest Content Manager V8.3. We have kept this chapter here for reference purposes.

To install and configure Content Manager V8.3, consult the latest publication manuals.
A.1 Installation overview for Windows (for V8.2 only)

This section covers the installation overview for Content Manager V8.2 on a Windows platform. It should be used in conjunction with the following product manuals:

- IBM Content Manager for Multiplatforms - Revised Installation Steps for Windows
- IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System, GC27-1332

**Attention:** This section covers the installation overview for Content Manager Version 8.2 on Windows only. It is kept here for reference purposes.

For an installation overview for Content Manager V8.3 on a Windows platform, consult the latest product manuals.

A.1.1 Content Manager prerequisites for Windows

Before Content Manager can be installed, various software prerequisite requirements must be met. Some items listed in Table A-1 need only be installed to enable certain Content Manager features (such as DB2 Net Search Extender for enabling Content Manager text search). Table A-1 contains a list of prerequisite products and verification procedures. You should review these carefully in order to ensure that you have the prerequisite software.

**Table A-1 Software prerequisite verification for Content Manager on Windows**

<table>
<thead>
<tr>
<th>Prerequisite product</th>
<th>How to check</th>
<th>An example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Windows NT SP6 or later</td>
<td>From a Windows Command Prompt: <code>winver</code></td>
<td>1. Version 4.0 (Build 1381: Service Pack 6)</td>
</tr>
<tr>
<td>2. Windows 2000 Server SP2 or later</td>
<td>From a Windows Command Prompt: <code>winver</code></td>
<td>2. Version 5.0 (Build 2195: Service Pack 2)</td>
</tr>
<tr>
<td>Java Development Kit V1.3 or greater</td>
<td>From a Windows Command Prompt: <code>java -fullversion</code></td>
<td>Version needs to read 1.3.1 or above (for example “1.4.1_01-b01”)</td>
</tr>
<tr>
<td>Prerequisite product</td>
<td>How to check</td>
<td>An example value</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>------------------</td>
</tr>
<tr>
<td>DB2 UDB EE V7.2 with fixpack 7 or greater (fixpack 10 or greater when CM 8.2 fixpack 1 is installed)</td>
<td>From a DB2 Command Window: db2level</td>
<td>Level needs to read “SQL07025” or greater with fixpack “WR21306” or greater. For CM 8.2 fixpack 1 the level should read “SQL07028” or greater with fixpack “WR21329” or greater.</td>
</tr>
<tr>
<td>DB2 UDB Enterprise Server Edition Version 8.1 with fixpack 1 or greater</td>
<td>From a DB2 Command Window: db2level</td>
<td>Level needs to read “SQL08010” or greater and read “DB2 v8.1.1.27” or greater. The fixpack information needs to read “FixPak “1”” or greater and list the fixpack level.</td>
</tr>
</tbody>
</table>
| DB2 Text Information Extender V7.2 with fixpack 1 or greater | From a DB2 Command Window: db2text start | 1. CTE0185  
2. CTE0001 Operation completed successfully |
| DB2 Net Search Extender V8.1 (required if you use DB2 Version 8.1 or greater) | From a DB2 Command Window, start the text search program: db2text start  
Then type: db2textlevel | CTE0350 Instance “DB2” uses DB2 Net Search Extender code release “tx9_81” with level identifier “tx9_26a” |
| Tivoli Storage Manager 32 bit API Client Version 4.2.1 or greater | C:\Program Files\tivoli\tsm\api\sample\dapismp  
or  
C:\Program Files\tivoli\tsm\client \api\sample\dapismp | API Library Version = 5.1.5.4 |
| Tivoli Storage Manager Server Version 4.2.1 or greater | Logon to the TSM Server Administration Web interface: http://<hostname>:1580  
Where <hostname> is the name of the TSM server. | The version appears on the top part of the web page. Version 5, Release 1, Level 5.4 |
<table>
<thead>
<tr>
<th>Prerequisite product</th>
<th>How to check</th>
<th>An example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. WebSphere Application Server AE v4.0.5 or greater</td>
<td>Check the product.xml file located by default in: C:\Program Files\WebSphere\AppServer\proper\com\ibm\webs phere</td>
<td>&lt;version&gt;4.0.5&lt;/version&gt;</td>
</tr>
<tr>
<td>2. WebSphere Application Server AES v4.0.5 or greater</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebSphere Application Server v5</td>
<td>Open the following file: BASE.product Located by default in: C:\Program Files\WebSphere\properties\version</td>
<td>&lt;version&gt;5.0.2&lt;/version&gt; &lt;build-info date=&quot;06/23/2003&quot; level=&quot;ptf2M0325.01&quot;/&gt;</td>
</tr>
<tr>
<td>Microsoft Visual Studio .NET Professional</td>
<td>From a Windows Command Prompt type: cl</td>
<td>Microsoft 32-bit C/C++ Optimizing Compiler Version 13.00.94966 for 80x86 Copyright (C)Microsoft Corporation 1984-2001. All rights reserved.</td>
</tr>
</tbody>
</table>

For instructions on upgrading your prerequisites to the required levels, please see the IBM manuals mentioned earlier, and the prerequisites’ own product documentation.

Before performing any prerequisite product installations, you should create the necessary operating system users that Content Manager requires.

### A.1.2 Set up users

You need to set up users prior to installing Content Manager.
Content Manager requires three operating system user IDs to be created before it can be installed:

- “Library Server administration” user ID (such as ICMADMIN) if you are installing a Library Server on this workstation. This user ID must be a member of the Windows administrators group (which is an easy way to make them part of the DB2 Admin Group).
- “Database connection” user ID (such as ICMCONCT) if you are installing a Library Server on this workstation. This should be a standard user ID with normal privileges.
- “Resource Manager administration” user ID (such as RMADMIN) if you are installing a Resource Manager on this workstation. This user ID must be a member of the Windows administrators group (which is an easy way to make them part of the DB2 Admin Group).

In addition, icmadmin and rmadmin need to have the following four user rights:

- Act as part of the operating system
- Create a token object
- Increase quotas
- Replace a process level token

Use the standard Windows administration tools in order to complete this step.

The user IDs and passwords created above are used during the Content Manager installation. Record the values you used here so you can easily retrieve them when installing Content Manager.

A.1.3 Configure Secure Sockets Layer (SSL) for IBM HTTP Server

If you have installed WebSphere Application Server on your workstation, you must configure Secure Sockets Layer (SSL) for IBM HTTP Server. This must be done to enable the Content Manager System Administration Client to communicate successfully with the Resource Manager.

For a detailed step-by-step guide to configuring SSL for IBM HTTP Server, see the IBM product manuals mentioned earlier.

To configure a secure network connection using SSL, the following five procedures should be followed:

1. Create a new key database (if one does not already exist) and a key.
2. Receive a server certificate from a certificate authority or create a self-signed server certificate using the IBM Key Management Utility (IKEYMAN).
3. Set up SSL using the IBM HTTP Administration Server.
4. Add port 443 to the WebSphere virtual host, host alias definitions within the default host (see 21.2.7, “Generating the Web server plug-in with SSL information for WebSphere Application Server” on page 584).

5. Test the server installation and configuration.

**A.1.4 WebSphere SSL configuration and localhost restriction**

When you install WebSphere, the Secure Sockets Layer (SSL) is configured with a “dummy” certificate. Before you go into production with Content Manager, make sure that the WebSphere plug-in certificate and Application Server certificate you are using is your own certificate or a self-signed certificate.

The WebSphere documentation describes how to configure SSL with your own certificate or self-signed certificate.

For security reasons, restrict the ports of your Resource Manager Application Server to “localhost”. Follow the instructions in the next section to update the host field.

**Important:** Port 9090 is usually reserved for the WebSphere administration port. You must decide if you want to restrict the WebSphere administration port to localhost when using WebSphere Application Server V5. For example, if you do this, you need to type http://localhost:9090/admin/ in order to access the WebSphere Application Server Administrative Console (for WebSphere version 5). This means that you can no longer administer WebSphere Application Server V5 via the Administrative Web Console from a remote machine.

**Restricting ports for WebSphere Application Server Version 5.0**

To restrict ports for WebSphere Application Server Version 5.0, do the following steps:

1. Be sure the administrative server is running and start the WebSphere Administration console.

2. Select Servers → Application Servers → <your application server name> → Web Container → HTTP Transports.

3. Update the host field for each port from “*” to “localhost”. Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.

4. For each port you changed in the step above for your Application Server, you must select Environment → Virtual Hosts → <default_host> → Host.
**Aliases** and update the specific port from * to localhost. *See the restriction for port 9090 in the grey box above.*

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

**Restricting ports for WebSphere Application Server Advanced Edition Version 4.0.5**

To restrict ports for WebSphere Application Server Advanced Edition Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server is running and start the WebSphere Administration console.

2. Select **Nodes** → <your Node name> → **Application Servers** → <your application server name> → **Services** → **Web Container Service** → **Transport**.

3. Update the transport host field for each port from * to localhost. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.*

4. For **each port you changed in the step above** for your Application Server, you must select **Virtual Hosts** → <default_host> → **General** and update the specific port from * to localhost.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

**Restricting ports for WebSphere Application Server Advanced Single Server Edition Version 4.0.5**

To restrict ports for WebSphere Application Server Advanced Single Server Edition Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server services are started and start the WebSphere Administration console.

2. Select **Nodes** → <your Node name> → **Application Servers** → **Default Server** → **Web Container** → **HTTP Transports**.

3. Update the host name field for each port from * to localhost. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.*
4. For each port you changed in the step above for your Application Server, you must select Virtual Hosts → <default_host> → Aliases and update the specific ports from * to localhost.
5. Regenerate the Web server plug-in.
6. Restart the HTTP server.
7. Restart WebSphere Application Server.

A.1.5 Installing Content Manager on Windows

Before installing Content Manager on Windows, you should make sure that all of the prerequisites you require are installed and working correctly.

**Important:** Make sure that you start the WebSphere Server Service (server1) before you begin the Content Manager installation procedure.

The user ID with which you install Content Manager must:

- Be defined locally.
- Belong to the Local Administrator’s group.
- Be one to eight characters in length.

**Important:** User IDs must also follow the rules outlined by the relational database manager you are using.

**Installing the Content Manager components on Windows**

To install Content Manager on Windows, do the following steps:

1. Insert the Content Manager CD into the CD-ROM drive of your workstation. The Content Manager installation launch pad opens. Select Install Product.
2. Select Next on the Welcome Panel.
3. Read the Content Manager license terms. Select Accept and select Next to continue the installation.
4. Choose the directory where the Content Manager program files to be installed. Select Next to continue the installation.
5. Select the components to install, then select Next.

**Library Server configuration**

6. Configure the Library Server: Complete this step and select Next. Skip this step if you are not installing the Library Server component at this time, and continue with step 8.
7. Configure Library Server options: Complete this step and select Next.
   Skip this step if the Library Server component is not being installed at this time.

Resource Manager configuration

8. Configure the Resource Manager server: Complete this step and select Next.
   Skip this step if you are not installing the Resource Manager component at this time, and continue with step 12.

9. Configure Resource Manager server options: Complete this step and select Next.
   Skip this step if the Resource Manager component is not being installed at this time.

10. Deploy the Resource Manager with WebSphere Application Server: complete this step and select Next.

11. Automatic deployment with WebSphere Application Server V5: This screen is shown in Figure A-1 and needs to be filled in with the correct values if global security has been enabled on WebSphere Application Server V5 (see 8.5, “WebSphere global security” on page 223); otherwise, the Resource Manager application will not be able to deploy on WebSphere. If global security has not been enabled, then the values you enter here are arbitrary, even if you are using WebSphere Application Server V5. Complete this step and select Next.
   Skip this step if you do not have WebSphere Application Server V5 installed on this workstation and continue with the next step.
System administration configuration

12. Configure the System Administration Client: Complete this step and select Next.

   Skip this step if the System Administration Client component is not being installed at this time.

13. Define the location of system configuration information: Complete this step and select Next.

Defining connections

14. Connect the Library Server to Resource Manager: Complete this step and select Next.

   Skip this step if you are not installing a Library Server or a Resource Manager at this time. Continue with step 20. Also, skip this step if you are only installing a Resource Manager, but not a Library Server at this time. In the later case, continue with step 16.

15. Connect a Library Server to Resource Manager: Complete this step and select Next.
Skip this step if the Library Server and the Resource Manager are being installed onto the same machine.

16. Connect a Resource Manager to a Library Server: Complete this step and select Next.

Skip this step if you are not installing a Resource Manager at this time, or if you are installing a Library Server and a Resource Manager on the same machine.

**LDAP integration**

17. Configure components for LDAP: Complete this step if you wish to integrate an LDAP directory with Content Manager. Select Next when this step has been completed.

18. Define an LDAP server: Skip this step if you are not integrating LDAP at this time, and go to step 20. Select Next when this step has been completed.

19. Configure LDAP server: Skip this step if you are not integrating LDAP at this time. Select Next when this step has been completed.

**Verify the install location and component selection**

20. Verify that the installation location and component selection is correct. If any parameters are incorrect, you can return to previous windows by using the Back buttons. Select Next to complete the installation.

21. The Start Copying Files window opens. You will see a message that the installation has been successful. Select Finish.

If you received a message during the installation, you can view it in the log.txt file in your `%ICMROOT%` directory (where `%ICMROOT%` is the directory where Content Manager is installed.)

### A.1.6 Installation verification

After the installation of a Content Manager System Administration Client, the First Steps launch pad appears. You can use it at this time, or you can come back to it later to verify the installation.

To use First Steps to verify the installation, do the following steps:

1. Select **Start** → **Programs** → **IBM Content Manager for Multiplatforms v8.2** → **First Steps.** The First Steps window opens:

2. Select **First Steps Information** to read the introduction to the First Steps process.

2. Select **Load Sample Data** to store the samples into the Content Manager database.
3. Select **Work with Sample Data**. The System Administration Client opens. You can use it to see how Content Manager makes use of the new data model to manage objects. Some examples of what you can do are as follows:

- You can open the item type Policy and go to the Attributes page:
  - Attributes and attribute groups appear to the left.
  - You can see that Policy is the name of the item type.
  - Insured and VIN are child components of Policy.
  - Address is an attribute group.
  - Policy_Number shows you an attribute that is free of a child component or an attribute group.

- You can explore the sample data for each object.
- You can create your own objects and add them to the sample data.
- You can delete users and re-create them.

For further information on verifying and troubleshooting your Content Manager installation, refer to the IBM manuals mentioned earlier in Section A.1, “Installation overview for Windows (for V8.2 only)” on page 602.

### A.2 Installation overview for AIX (for V8.2 only)

This section covers the installation overview for Content Manager V8.2 on an AIX platform, and should be used in conjunction with the following product manuals:

- *IBM Content Manager for Multiplatforms - Revised Installation Steps for AIX*
- *IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332

**Attention:** This section covers the installation overview for Content Manager Version 8.2 on the AIX platform only. It is kept here for reference purposes.

For installation overview for Content Manager V8.3 on an AIX platform, consult the latest product manuals.

#### A.2.1 Content Manager prerequisites for AIX

Before Content Manager can be installed, various software prerequisite requirements must be met. Some items listed in Table A-2 need only be installed to enable certain Content Manager features (such as DB2 Net Search Extender for enabling Content Manager text search). Table A-2 contains a list of prerequisite products and verification procedures. You should review these carefully in order to ensure that you have the prerequisite software.
<table>
<thead>
<tr>
<th>Prerequisite product</th>
<th>How to check</th>
<th>An example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. AIX 4.3.3 ML 9 or higher + APAR IY19277</td>
<td>oslevel -r</td>
<td>1. 4330-09 or higher</td>
</tr>
<tr>
<td>2. AIX 5.1 ML 1 or higher</td>
<td></td>
<td>2. 5100-01 or higher</td>
</tr>
<tr>
<td>IBM VisualAge® C++ Professional Batch Compiler version 5.0.2.0 (on AIX 5.1 you need to install ptfs IY18426 and IY23677)</td>
<td>lslpp -L vacpp.cmp.batch</td>
<td>Level: 5.0.2.0 or higher (for AIX 5.1 needs to read 5.0.2.2 or higher)</td>
</tr>
<tr>
<td>DB2 UDB EE v7.2 with fixpack 7 or greater (fixpack 10 or greater when CM 8.2 fixpack 1 is installed)</td>
<td>db2level</td>
<td>Level needs to read “SQL07025” or greater with fixpack “U484480” or greater. For CM 8.2 fixpack 1 the level should read “SQL07028” or greater with fixpack “U488490” or greater.</td>
</tr>
<tr>
<td>DB2 UDB Enterprise Server Edition Version 8.1 with fixpack 1 or greater</td>
<td>db2level</td>
<td>Level needs to read “SQL08010” or greater and read “DB2 v8.1.1.27” or greater. The fixpack information needs to read “FixPak “1”” or greater and list the fixpack level.</td>
</tr>
<tr>
<td>DB2 Text Information Extender V7.2 with fixpack 1 or greater</td>
<td>lslpp -l</td>
<td>grep db2tie</td>
</tr>
<tr>
<td>DB2 Net Search Extender v8.1 (required if you use DB2 Version 8.1 or greater)</td>
<td>lslpp -l</td>
<td>grep db2nse</td>
</tr>
<tr>
<td>Tivoli Storage Manager 32 bit API Client Version 4.2.1 or greater</td>
<td>/usr/tivoli/tsm/client/api/samprun/dapismp or /usr/tivoli/tsm/client/api/bin/sample/dapismp</td>
<td>API Library Version = 5.1.5.4</td>
</tr>
<tr>
<td>Prerequisite product</td>
<td>How to check</td>
<td>An example value</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Tivoli Storage Manager Server Version 4.2.1 or greater                               | Logon to the TSM Server Administration web interface: \texttt{http://<hostname>:1580}  
Where \texttt{<hostname>} is the name of the TSM server. | The version appears on the top part of the web page. 
Version 5, Release 1, Level 5.4                                                    |
| Java 1.3.1 or greater                                                                | \texttt{java -version}                                                        | \texttt{java version “1.3.1”} or greater                                         |
| 1. WebSphere Application Server AE V4.0.5 or greater                                 | \texttt{grep “<version>” \slash/usr/WebSphere/AppServer/properties/com/ibm/websphere/product.xml} | `<version>4.0.5</version>`                                                     |
| 2. WebSphere Application Server AES V4.0.5 or greater                                | \texttt{grep “<version>” \slash/usr/WebSphere/AppServer/properties/com/ibm/websphere/product.xml} | `<version>5.0.2</version>`                                                     |
| WebSphere Application Server V5                                                      | Open the following file: \texttt{BASE.product} \texttt{Located by default in:} \texttt{/usr/WebSphere/AppServer/properties/version/} | `<version>5.0.2</version>`                                                     |

For instructions on upgrading your prerequisites to the required levels, please see the IBM manuals mentioned in A.2, “Installation overview for AIX (for V8.2 only)” on page 612, and the prerequisites’ own product documentation.
A.2.2 Set up users

You need to set up users prior to installing Content Manager. This includes doing the following tasks:

- Create user IDs.
- Give icmadmin and rmadmin DB2 Admin privileges.
- Update the .profiles for the new user IDs.
- Update the DB2 instance profile.env file.
- Create a userprofile file for Content Manager environment settings.

Create user IDs

Content Manager requires three operating system user IDs to be created before it can be installed:

- “Library Server administration” user ID (such as ICMADMIN) if you are installing a Library Server on this workstation. This user ID must be a member of the DB2 Admin group (this step can be done after installing DB2).
- “Database connection” user ID (such as ICMCONCT) if you are installing a Library Server on this workstation. This should be a standard user ID with normal privileges.
- “Resource Manager administration” user ID (such as RMADMIN) if you are installing a Resource Manager on this workstation. This user ID must be a member of the DB2 Admin group (this step can be done after installing DB2).
These user IDs can be created using `smitty`, and should be given a password. Perform an initial login using each of these user IDs. You will be asked to change the password you have set. You can then simply enter the password you originally set again.

The user IDs and passwords created above are used during the Content Manager installation. Record the values you used here so you can easily retrieve them when installing Content Manager.

**Give icmadmin and rmadmin DB2 Admin privileges**

These user IDs can be created before or after the DB2 installation. As stated above, the icmadmin user ID and the rmadmin user ID need to be a part of the DB2 Admin group.

This can be achieved by changing the primary group of these users to the instance owners group that was created during the DB2 installation. The defaults during the installation create a group called `db2iadm1` for the instance owner (`db2inst1` by default). Change the primary groups for icmadmin and rmadmin to `db2iadm1` (if the defaults are taken), any users added to this group will have DB2 Admin authority over databases created within the instance. You can use `smitty` to perform this procedure.

The root user also needs to be added to the `db2` instance owners primary group (do not change root's primary group to `db2iadm1`, just include `db2iadm1` within root's group set). Using the example names and groups above, icmadmin, rmadmin and root should all be members of the group `db2iadm1`.

If you have two instances, one for the Library Server database and one for the Resource Manager database (as recommended for a production Content Manager system), then you would change icmadmin's primary group to the instance owners primary group for the Library Server instance, and do the same for rmadmin using the Resource Manager instance owners primary group.

**Update the .profiles for the new user IDs**

Add the following line to `/home/icmadmin/.profile` and `/home/rmadmin/.profile` files (this must be done after the DB2 installation):

```
./home/db2inst1/sql/lib/db2profile
```

Notice the space between the period (.) and the first slash (/). This will establish the DB2 environment associate the users with the `db2inst1` DB2 instance (adjust accordingly if your instance is not called `db2inst1`).

If you are using DB2 V7.2, you should also put the following line in the .profile's of the icmadmin and rmadmin users:

```
./home/db2inst1/sql/lib/java12/usejdbc2
```
Notice the space between the period (.) and the first slash (/).

**Update the DB2 instance profile.env file**

If the data is not already in the file, add the following lines to the /home/db2inst1/sqllib/profile.env file (if profile.env does not exist, create it):

```
DB2ENVLIST='LIBPATH ICMROOT ICMDLL ICMCOMP EXTSHM CMCOMMON'
DB2COMM='tcpip'
DB2AUTOSTART='TRUE'
```

**Create a userprofile file for Content Manager environment settings**

Create a file or update the file called: home/db2inst1/sqllib/userprofile to contain the following data:

```
ICMROOT=/usr/lpp/icm
ICMDLL=/home/db2fenc1
ICMCOMP=/usr/vacpp/bin
CMCOMMON=/usr/lpp/cmb/cmgmt
EXTSHM=ON
PATH=$PATH:$ICMROOT/bin/DB2
LIBPATH=$ICMROOT/lib:$ICMROOT/inso:$LIBPATH
export ICMROOT ICMCMDL ICMCOMP CMCOMMON EXTSHM PATH LIBPATH
```

Do not modify /home/db2inst1/sqllib/db2profile, since this file can be overwritten by the application of a DB2 fixpack. Instead, put any necessary modifications in userprofile. When db2profile is invoked, it runs userprofile. When it runs userprofile, it causes all settings added to it to be set for users in db2profile. This action establishes the DB2 environment with db2profile.

**Important:** You must restart DB2 for the changes that you made to take effect.

### A.2.3 Configure Secure Sockets Layer (SSL) for IBM HTTP Server

If you have installed WebSphere Application Server on your workstation, you must configure Secure Sockets Layer (SSL) for IBM HTTP Server. This must be done to enable the Content Manager System Administration Client to communicate successfully with the Resource Manager.

For a detailed step-by-step guide to configuring SSL for IBM HTTP Server, see the IBM product manuals mentioned above in Section A.2, “Installation overview for AIX (for V8.2 only)” on page 612.

To configure a secure network connection using SSL, follow these five procedures:

1. Create a new key database (if one does not already exist) and a key.
2. Receive a server certificate from a certificate authority or create a self-signed server certificate using the IBM Key Management Utility (IKEYMAN).

3. Set up SSL using the IBM HTTP Administration Server or editing the /usr/IBMHttpServer/conf/httpd.conf file directly (take a backup of this file before you begin to edit it).

4. Add port 443 to the WebSphere virtual host, host alias definitions within the default host (see Section 21.2.7, “Generating the Web server plug-in with SSL information for WebSphere Application Server” on page 584).

5. Test the server installation and configuration.

### A.2.4 WebSphere SSL configuration and localhost restriction

When you install WebSphere, the Secure Sockets Layer (SSL) is configured with a “dummy” certificate. Before you go into production with Content Manager, make sure that the WebSphere plug-in certificate and Application Server certificate you are using is your own certificate or a self-signed certificate.

The WebSphere documentation describes how to configure SSL with your own certificate or self-signed certificate.

For security reasons, restrict the ports of your Resource Manager Application Server to “localhost”. Follow the instructions in the next section to update the host field.

**Important:** Port 9090 is usually reserved for the WebSphere administration port. You must decide if you want to restrict the WebSphere administration port to localhost. For example, if you do this, you need to type http://localhost:9090/admin/ in order to access the WebSphere Application Server Administrative Console (for WebSphere version 5). This means you can no longer administer WebSphere Application Server via the Administrative Web Console from a remote machine.

**Restricting ports for WebSphere Application Server Version 5.0**

To restrict ports for WebSphere Application Server Version 5.0, do the following steps:

1. Be sure the administrative server is running and start the WebSphere Administration console.

2. Select **Servers → Application Servers → <your application server name> → Web Container → HTTP Transports.**

3. Update the host field for each port from “*” to “localhost”. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If
they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.

4. For each port you changed in the step above for your Application Server, you must select Environment → Virtual Hosts → <default_host> → Host Aliases and update the specific port from * to localhost. See the restriction for port 9090 in the grey box above.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

Restricting ports for WebSphere Application Server Advanced Edition Version 4.0.5

To restrict ports for WebSphere Application Server Advanced Edition Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server is running and start the WebSphere Administration console.

2. Select Nodes → <your Node name> → Application Servers → <your application server name> → Services → Web Container Service → Transport.

3. Update the transport host field for each port from * to localhost. Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.

4. For each port you changed in the step above for your Application Server, you must select Virtual Hosts → <default_host> → General and update the specific port from * to localhost.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

Restricting ports for WebSphere Application Server Advanced Single Server Edition Version 4.0.5

To restrict ports for WebSphere Application Server Advanced Single Server Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server services are started and start the WebSphere Administration console.

2. Select Nodes → <your Node name> → Application Servers → Default Server → Web Container → HTTP Transports.
3. Update the host name field for each port from * to localhost. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.*

4. For each port you changed in the step above for your Application Server, you must select Virtual Hosts → <default_host> → Aliases and update the specific ports from * to localhost.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

### A.2.5 Installing Content Manager on AIX

*Before* installing Content Manager on AIX, you should make sure that all of the prerequisites you require are installed and working correctly.

**Important:** Make sure that you *start* the WebSphere Server Service (server1) before you begin the Content Manager installation procedure.

---

**Create a staging directory for the Resource Manager**

During the installation, you are prompted to provide the staging area directory and its mount point. The installation program assumes you have already created this directory. For example:

```bash
mkdir /home/RM1staging
```

**Create an object storage file-system for the Resource Manager**

During the Content Manager installation, you are prompted to provide the application storage mount point. The installation program assumes that you have already created a file-system and mount point. It also assumes that you have already mounted the file-system.

Refer to the AIX help for commands to create a file-system. The command to mount the file-system is similar to the following command:

```bash
mount /home/lbosdata
```

**Establish the database environment**

It is *very important* that you establish the DB2 environment for Content Manager by following the instructions to set up the userprofile in the sqllib directory (refer to “Create a userprofile file for Content Manager environment settings” on page 617). You must then run db2profile (run this command as root just before
you install Content Manager) as this sets the PATH and CLASSPATH and also identifies the DB2 instance that Content Manager uses:

  . /home/db2inst1/sql1ib/db2profile

**Important:** Do not forget to establish the database environment. If you forget to do this, Content Manager will not be installed successfully.

**Installing the Content Manager components on AIX**

To install Content Manager on AIX, do the following steps:

1. Insert the Content Manager CD into the CD-ROM drive of your workstation.
2. Log in as root (or a user with root authority).
3. Mount the Content Manager CD-ROM, for example:

   ```bash
   mount -rv cdrfs /dev/cd0 /cdrom
   ```

4. Change to the directory where the CD-ROM is mounted.
5. Start the installation wizard by entering the following command:

   ```bash
   setup.exe
   ```

6. Select **Next** on the Welcome Panel.
7. Read the Content Manager license terms. Select **Accept** and select **Next** to continue the installation.
8. Choose the directory where the Content Manager program files will be installed. Select **Next** to continue the installation.
9. Select the components to install, then select **Next**.

**Library Server configuration**

10. Configure the Library Server: Complete this step and select **Next**.
    
    Skip this step if you are not installing the Library Server component at this time, and continue with step 12.

11. Configure Library Server options: Complete this step and select **Next**.
    
    Skip this step if the Library Server component is not being installed at this time.

**Resource Manager configuration**

12. Configure the Resource Manager server: Complete this step and select **Next**.
    
    Skip this step if you are not installing the Resource Manager component at this time, and continue with step 16.

13. Configure Resource Manager server options: Complete this step and select **Next**.
Skip this step if the Resource Manager component is not being installed at this time.

14. Deploy the Resource Manager with WebSphere Application Server: Complete this step and select **Next**.

15. Automatic deployment of Resource Manager with WebSphere Application Server V5: This screen is shown in Figure A-2 and needs to be filled in with the correct values if global security has been enabled on WebSphere Application Server V5 (see 8.5, “WebSphere global security” on page 223); otherwise, the Resource Manager application will not be able to deploy on WebSphere. If global security has not been enabled, then the values you enter here are arbitrary, even if you are using WebSphere Application Server V5. Complete this step and select **Next**.

Skip this step if you do not have WebSphere Application Server V5 installed on this workstation and continue with the next step.

![Figure A-2 Automatic deployment with WebSphere Application Server v5](image)

**Defining connections**

16. Connect the Library Server to Resource Manager: Complete this step and select **Next**.

Skip this step if you are not installing a Library Server or a Resource Manager at this time and then continue with step 22. Also, skip if you are only installing
a Resource Manager, *but not* a Library Server at this time. For the later case, continue with step 18.

17. Connect a Library Server to Resource Manager: Complete this step and select **Next**.

Skip this step if the Library Server and the Resource Manager are being installed onto the same machine.

18. Connect a Resource Manager to a Library Server: Complete this step and select **Next**.

Skip this step if you are not installing a Resource Manager at this time, or if you are installing a Library Server and a Resource Manager on the same machine.

**LDAP integration**

19. Configure components for LDAP: Complete this step if you wish to integrate an LDAP directory with Content Manager. Select **Next** when this step has been completed.

20. Define an LDAP server: Skip this step if you are not integrating LDAP at this time; otherwise, complete this step and select **Next**.

21. Configure LDAP server: Skip this step if you are not integrating LDAP at this time; otherwise, complete this step and select **Next**.

**Verify the install location and component selection**

22. Verify that the installation location and component selection is correct. If any parameters are incorrect, you can return to previous windows by using the **Back** buttons. Select **Next** to complete the installation.

23. The Start Copying Files window opens. You will see a message that the installation has been successful. Select **Finish**.

You can view the installation logs at the following location:

/\usr/lpp/icm/logs

**A.2.6 Installation verification**

After the installation is complete, you can go to a Windows workstation that has the Content Manager System Administration Client installed to verify that the installation is successful by running First Steps. See Section A.1.6, “Installation verification” on page 611 for instructions.

For further information on verifying and troubleshooting your Content Manager installation, refer to the IBM manuals mentioned earlier in Section A.2, “Installation overview for AIX (for V8.2 only)” on page 612.
A.3 Installation overview for Sun Solaris (for V8.2 only)

This section covers the installation overview for Content Manager V8.2 on a Sun Solaris platform. It should be used in conjunction with the following product manuals:

- IBM Content Manager for Multiplatforms - Revised Installation Steps for Sun Solaris
- IBM Content Manager for Multiplatforms - Planning and Installing Your Content Management System, GC27-1332

**Attention:** This section covers the installation overview for Content Manager Version 8.2 on Sun Solaris only. It is kept here for reference purposes.

For installation overview for Content Manager V8.3 on a Sun Solaris platform, consult the latest product manuals.

A.3.1 Content Manager prerequisites for Sun Solaris

Before Content Manager can be installed, various software prerequisite requirements must be met. Some items listed in Table A-3 need only be installed to enable certain Content Manager features (such as DB2 Net Search Extender for enabling Content Manager text search). Table A-3 contains a list of prerequisite products and verification procedures. You should review these carefully in order to ensure that you have the prerequisite software.

<table>
<thead>
<tr>
<th>Prerequisite product</th>
<th>How to check</th>
<th>An example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris Version 2.8 or later</td>
<td><code>uname -r</code></td>
<td>Level#: 5.8</td>
</tr>
<tr>
<td>Solaris patch 108528</td>
<td>`showrev</td>
<td>grep version`</td>
</tr>
<tr>
<td>Sun Forte™ C++ compiler V6.1 Enterprise Edition for Solaris with Update 1 or greater</td>
<td>`pkginfo -1 SP0vws</td>
<td>grep VERSION`</td>
</tr>
<tr>
<td>DB2 UDB Enterprise Server Edition Version 8.1 Solaris 32 &amp; 64 bit with fixpack 1 or greater</td>
<td><code>db2level</code></td>
<td>Level needs to read “SQL08010” or greater and read “DB2 v8.1.1.27” or greater. The fixpack information needs to read “FixPak “1”” or greater and list the fixpack level.</td>
</tr>
</tbody>
</table>
### Appendix A. Installation and configuration overview (for V8.2 only)

For instructions on upgrading your prerequisites to the required levels, please see the IBM manuals mentioned in A.3, “Installation overview for Sun Solaris (for V8.2 only)” on page 624, and the prerequisites’ own product documentation.

#### A.3.2 Set up users

You need to set up users prior to installing Content Manager. This includes doing the following tasks:

<table>
<thead>
<tr>
<th>Prerequisite product</th>
<th>How to check</th>
<th>An example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Net Search Extender V8.1</td>
<td>pkginfo -l db2nse81</td>
<td>VERSION: 8.1.0.0</td>
</tr>
<tr>
<td>Tivoli Storage Manager 32 bit API Client Version 4.2.1 or greater</td>
<td>/opt/tivoli/tsm/client/api/samprun/dapismp</td>
<td>API Library Version = 5.1.5.4</td>
</tr>
<tr>
<td>Tivoli Storage Manager Server Version 5.1 or greater</td>
<td>Logon to the TSM Server Administration web interface: http://&lt;hostname&gt;:1580</td>
<td>The version appears on the top part of the web page. Version 5, Release 1, Level 5.4</td>
</tr>
<tr>
<td>Java 1.3.1 or greater</td>
<td>java -version</td>
<td>java version “1.3.1” or greater</td>
</tr>
<tr>
<td>1. WebSphere Application Server AE V4.0.5 or greater</td>
<td>grep /version /opt/WebSphere/AppServer/properties/com/ibm/websphere/product.xml</td>
<td>&lt;version&gt;5.0.2&lt;/version&gt; &lt;build-info date=&quot;06/23/2003&quot; level=&quot;ptf2M0325.01&quot;/&gt;</td>
</tr>
<tr>
<td>2. WebSphere Application Server AES V4.0.5 or greater</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. WebSphere Application Server V5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create user IDs.
Give icmadmin and rmadmin DB2 Admin privileges.
Update the .profiles for the new user IDs.
Update the DB2 instance profile.env file.
Create a userprofile file for Content Manager environment settings.

Create user IDs
Content Manager requires three operating system user IDs to be created before it can be installed:

- “Library Server administration” user ID (such as ICMADMIN) if you are installing a Library Server on this workstation. This user ID must be a member of the DB2 Admin group (this step can be done after installing DB2).
- “Database connection” user ID (such as ICMCONCT) if you are installing a Library Server on this workstation. This should be a standard user ID with normal privileges.
- “Resource Manager administration” user ID (such as RMADMIN) if you are installing a Resource Manager on this workstation. This user ID must be a member of the DB2 Admin group (this step can be done after installing DB2).

Perform an initial login using each of these user IDs. You will be asked to change the password you have set; you can then simply enter the password you originally set, again.

The user IDs and passwords created above are used during the Content Manager installation. Record the values you used here so you can easily retrieve them when installing Content Manager.

Give icmadmin and rmadmin DB2 Admin privileges
These user IDs can be created before or after the DB2 installation. As stated before, the icmadmin user ID and the rmadmin user ID need to be a part of the DB2 Admin group.

This can be achieved by changing the primary group of these users to the instance owners group that was created during the DB2 installation. The defaults during the installation create a group called db2iadm1 for the instance owner (db2inst1 by default). Change the primary groups for icmadmin and rmadmin to db2iadm1 (if the defaults are taken), any users added to this group will have DB2 Admin authority over databases created within the instance.

The root user also needs to be added to the db2 instance owners primary group (do not change root’s primary group to db2iadm1, just include db2iadm1 within root’s group set). Using the example names and groups above, icmadmin, rmadmin and root should all be members of the group db2iadm1.
If you have two instances, one for the Library Server database and one for the Resource Manager database (as recommended for a production Content Manager system), then you would change icmadmin’s primary group to the instance owners primary group for the Library Server instance, and do the same for rmadmin using the Resource Manager instance owners primary group.

**Update the .profiles for the new user IDs**

Add the following lines to `/export/home/icmadmin/.profile` and `/export/home/rmadmin/.profile` files:

```
ICMROOT=/opt/IBMicm
ICMDLL=/export/home/db2fence1
ICMCOMP=/opt/SUNWspro/bin
CMCOMMON=/opt/IBMcmb/cmgmt
PATH=$PATH:$ICMROOT/bin/DB2
LD_LIBRARY_PATH=$ICMROOT/lib:$ICMROOT/inso:/opt/SUNWspro/lib:/usr/lib:$LD_LIBRARY_PATH
export ICMROOT ICMDLL ICMCOMP CMCOMMON PATH LD_LIBRARY_PATH
```

Where:
- ICMROOT is the Content Manager product install location.
- ICMDLL is the DB2 fence location (This is set to home of DB2fence because the fenced ID creates the DLL dynamically at run time).
- ICMCOMP is the Forte C++ compiler location.
- CMCOMMON is the shared area for Content Manager configuration files.

Notice the space between the period (.) and the first slash (/) on the last entry. This line establishes the DB2 environment associating the users with the db2inst1 DB2 instance.

Also add the following lines to the same file, if they are not already there:

```
if [[-e $DB2INSTANCE HOME/sqllib/db2profile]]then
 . $DB2INSTANCE HOME/sqllib/db2profile
fi
```

**Update the DB2 instance profile.env file**

If the data is not already in the file, add the following lines to the `/export/home/db2inst1/sqllib/profile.env` file (if profile.env does not exist, create it):

```
DB2ENVLIST='LD_LIBRARY_PATH ICMROOT ICMDLL ICMCOMP CMCOMMON'
DB2COMM='tcpip'
```
Create a userprofile file for Content Manager environment settings

Create a file or update the file: /export/home/db2inst1/sqllib/userprofile containing the following information:

```
ICMROOT=/opt/IBMicm
ICMDLL=/export/home/db2fenc1
ICMCOMP=/opt/SUNWspro/bin
CMCOMMON=/opt/IBMcmb/cmgmt
PATH=$PATH:$ICMROOT/bin/DB2
LD_LIBRARY_PATH=$ICMROOT/lib:$ICMROOT/inso:$LD_LIBRARY_PATH
export ICMROOT ICMCMDLL ICMCOMP CMCOMMON PATH LD_LIBRARY_PATH
```

Do not modify /export/home/db2inst1/sqllib/db2profile, since this file can be overwritten by the application of a DB2 fixpack. Instead, put any necessary modifications in userprofile. When db2profile is invoked, it runs userprofile. When it runs userprofile, it causes all settings added to it to be set for users in db2profile. This action establishes the DB2 environment with db2profile.

**Important:** You must restart DB2 for the changes that you made to take effect.

A.3.3 Configure Secure Sockets Layer (SSL) for IBM HTTP Server

If you have installed WebSphere Application Server on your workstation, you must configure Secure Sockets Layer (SSL) for IBM HTTP Server. This must be done to enable the Content Manager System Administration Client to communicate successfully with the Resource Manager.

For a detailed step-by-step guide to configuring SSL for IBM HTTP Server, refer to the IBM product manuals mentioned earlier in A.3, “Installation overview for Sun Solaris (for V8.2 only)” on page 624.

To configure a secure network connection using SSL, the following five procedures should be followed:

1. Create a new key database (if one does not already exist) and a key.
2. Receive a server certificate from a certificate authority or create a self-signed server certificate using the IBM Key Management Utility (IKEYMAN).
3. Set up SSL using the IBM HTTP Administration Server or editing the /usr/IBMHttpServer/conf/httpd.conf file directly (take a backup of this file before you begin to edit it).
4. Add port 443 to the WebSphere virtual host, host alias definitions within the default host (see 21.2.7, “Generating the Web server plug-in with SSL information for WebSphere Application Server” on page 584).
5. Test the server installation and configuration.
A.3.4 WebSphere SSL configuration and localhost restriction

When you install WebSphere, the Secure Sockets Layer (SSL) is configured with a “dummy” certificate. Before you go into production with Content Manager, make sure that the WebSphere plug-in certificate and Application Server certificate you are using is your own certificate or a self-signed certificate.

The WebSphere documentation describes how to configure SSL with your own certificate or self-signed certificate.

For security reasons, restrict the ports of your Resource Manager Application Server to “localhost”. Follow the instructions in the next section to update the host field.

Important: Port 9090 is usually reserved for the WebSphere administration port. You must decide if you want to restrict the WebSphere administration port to localhost. For example, if you do this, you need to type http://localhost:9090/admin/ in order to access the WebSphere Application Server Administrative Console (for WebSphere Version 5). This means you can no longer administer WebSphere Application Server via the Administrative Web Console from a remote machine.

Restricting ports for WebSphere Application Server Version 5.0

To restrict ports for WebSphere Application Server Version 5.0, do the following steps:

1. Be sure the administrative server is running and start the WebSphere Administration console.

2. Select Servers → Application Servers → <your application server name> → Web Container → HTTP Transports.

3. Update the host field for each port from “*” to “localhost”. Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.

4. For each port you changed in the step above for your Application Server, you must select Environment → Virtual Hosts → <default_host> → Host Aliases and update the specific port from * to localhost. See the restriction for port 9090 in the grey box above.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.
Restricting ports for WebSphere Application Server Advanced Edition Version 4.0.5

To restrict ports for WebSphere Application Server Advanced Edition Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server is running and start the WebSphere Administration console.

2. Select **Nodes** → *<your Node name>* → **Application Servers** → *<your application server name>* → **Services** → **Web Container Service** → **Transport**.

3. Update the transport host field for each port from * to localhost. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.*

4. For each port you changed in the step above for your Application Server, you must select **Virtual Hosts** → *<default_host>* → **General** and update the specific port from * to localhost.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.

Restricting ports for WebSphere Application Server Advanced Single Server Edition Version 4.0.5

To restrict ports for WebSphere Application Server Advanced Single Server Edition Version 4.0.5, do the following steps:

1. Be sure the WebSphere Application Server services are started and start the WebSphere Administration console.

2. Select **Nodes** → *<your Node name>* → **Application Servers** → **Default Server** → **Web Container** → **HTTP Transports**.

3. Update the host name field for each port from * to localhost. *Only do this if your HTTP server and WebSphere Application Server are on the same machine. If they are not, then instead of localhost, enter the fully qualified hostname of the HTTP server.*

4. For each port you changed in the step above for your Application Server, you must select **Virtual Hosts** → *<default_host>* → **Aliases** and update the specific ports from * to localhost.

5. Regenerate the Web server plug-in.

6. Restart the HTTP server.

7. Restart WebSphere Application Server.
A.3.5 Installing Content Manager on Sun Solaris

Before installing Content Manager on Sun Solaris, you should make sure that all of the prerequisites you require are installed and working correctly.

Important: Make sure that you start the WebSphere Server Service (server1) before you begin the Content Manager installation procedure.

Create a staging directory for the Resource Manager
During the installation, you are prompted to provide the staging area directory and its mount point. The installation program assumes you have already created this directory. For example:

```
mkdir /export/home/RM1staging
```

Create an object storage file-system for the Resource Manager
During the Content Manager installation, you are prompted to provide the application storage mount point. The installation program assumes that you have already created a file-system and mount point. It also assumes that you have already mounted the file-system.

Refer to the Sun Solaris help for commands to create a file-system. The command to mount the file-system is similar to the following command:

```
mount /export/home/lbosdata
```

Establish the database environment
It is very important that you establish the DB2 environment for Content Manager by following the instructions to set up the userprofile in the sqllib directory (refer to “Create a userprofile file for Content Manager environment settings” on page 628). You must then run db2profile (run this command as root just before you install Content Manager) as this sets the PATH and CLASSPATH and also identifies the DB2 instance that Content Manager uses:

```
./export/home/db2inst1/sqllib/db2profile
```

Important: Do not forget to establish the database environment. If you forget to do this, Content Manager will not be installed successfully.
Installing the Content Manager components on Sun Solaris

To install Content Manager on Sun Solaris, do the following steps:

1. Shut down any DB2 applications, then stop (db2stop) and start (db2start) DB2 with one of the following procedures (make sure all of the steps in A.3.1, “Content Manager prerequisites for Sun Solaris” on page 624 have been followed):
   - If you are installing a Library Server on this machine, login as a the Library Server administrator (for example: icmadmin) to shut down any open DB2 applications, then stop and start DB2 with the same user ID.
   - If you are only installing a Resource Manager on this machine, login as a the Resource Manager administrator (for example: rmadmin) to shut down any open DB2 applications, then stop and start DB2 with the same user ID.
   - If you are installing both a Library Server and a Resource Manager, and if they are being installed against separate DB2 instances, you need to shut down DB2 applications, then stop and start DB2 using both administrator IDs (for example: icmadmin and rmadmin).

2. Stop the IBM HTTP Server service by entering:
   `/opt/IBMHttpServer/bin/apachectl stop`

3. Log in as root by entering:
   `su - root`

4. Make sure that your Java JRE Version 1.3.1 (enter java -version) is in the PATH, for example:
   `$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH`

5. Install Content Manager by inserting the Content Manager CD into the CD-ROM drive of your workstation.

6. Mount the Content Manager CD-ROM, for example:
   `mount /cdrom`

7. Change to the English directory where the CD-ROM is mounted.

8. Set your DISPLAY by entering:
   `DISPLAY=hostname:0.0`

9. Start the installation wizard by entering the following command:
   `setup.exe`

10. Select Next on the Welcome Panel.

11. Read the Content Manager license terms. Select Accept and then select Next to continue the installation.
12. Choose the directory where the Content Manager program files will be installed. Select Next to continue the installation.

13. Select the components to install, then select Next.

**Library Server configuration**

14. Configure the Library Server: Complete this step and select Next.

   Skip this step if you are not installing the Library Server component at this time, and continue with step 12.

15. Configure Library Server options: Complete this step and select Next.

   Skip this step if the Library Server component is not being installed at this time.

**Resource Manager configuration**

16. Configure the Resource Manager server: Complete this step and select Next.

   Skip this step if you are not installing the Resource Manager component at this time, and continue with step 16.

17. Configure Resource Manager server options: Complete this step and select Next.

   Skip this step if the Resource Manager component is not being installed at this time.

18. Deploy the Resource Manager with WebSphere Application Server: Complete this step and select Next.

19. Automatic deployment with WebSphere Application Server V5: This screen is shown in Figure A-3 and needs to be filled in with the correct values if global security has been enabled on WebSphere Application Server V5 (see 8.5, “WebSphere global security” on page 223); otherwise, the Resource Manager application will not be able to deploy on WebSphere. If global security has not been enabled, then the values you enter here are arbitrary, even if you are using WebSphere Application Server V5. Complete this step and select Next.

   Skip this step if you do not have WebSphere Application Server V5 installed on this workstation and continue with the next step.
Defining connections

20. Connect the Library Server to Resource Manager: Complete this step and select Next.

Skip this step if you are not installing a Library Server or a Resource Manager at this time and then continue with step 26. Also, skip if you are only installing a Resource Manager, but not a Library Server at this time. For the later case, continue with step 22.


Skip this step if the Library Server and the Resource Manager are being installed onto the same machine.

22. Connect a Resource Manager to a Library Server: Complete this step and select Next.

Skip this step if you are not installing a Resource Manager at this time, or if you are installing a Library Server and a Resource Manager on the same machine.
LDAP integration

23. Configure components for LDAP: Complete this step if you wish to integrate an LDAP directory with Content Manager. Select Next when this step has been completed.

24. Define an LDAP server: Skip this step if you are not integrating LDAP at this time; otherwise, complete this step and select Next.

25. Configure LDAP server: Skip this step if you are not integrating LDAP at this time; otherwise, complete this step and select Next.

Verify the install location and component selection

26. Verify that the installation location and component selection is correct. If any parameters are incorrect, you can return to previous windows by using the Back buttons. Select Next to complete the installation.

27. The Start Copying Files window opens. You will see a message that the installation has been successful. Select Finish.

You can view the installation logs at the following location:

/opt/IBMicm/logs

A.3.6 Installation verification

After the installation is complete, you can go to a Windows workstation that has the Content Manager System Administration Client installed to verify that the installation is successful by running First Steps. See A.1.6, “Installation verification” on page 611 for instructions.

For further information on verifying and troubleshooting your Content Manager installation refer to the IBM manuals mentioned earlier in A.3, “Installation overview for Sun Solaris (for V8.2 only)” on page 624.

A.4 Installation process for z/OS (for V8.2 only)

The installation of Content Manager V8.2 for z/OS is done via SMP/E and is not covered in this redbook. For information regarding SMP/E, refer to the appropriate documentations.

In this section, we describe how to implement and customize Content Manager V8.2 for z/OS. We do the implementation and customization as described in the IBM product manual, Planing and Installing Your Content Management System for z/OS, GC18-7698.
A.4.1 Preparing and planning for the installation

As usual, preparing for the installation is the most important thing during the installation and implementation process. Some tasks need to be done as pre-installation steps. This is covered in chapters 4 and 5 of the IBM product manual, *Planing and Installing Your Content Management System for z/OS*, GC18-7698. In addition to the information given to you in the manual, you should also have the following publications available for reference:

- *Program Directory for IBM DB2 Content Manager Library Server for z/OS*, GI10-8530
- *Program Directory for IBM DB2 Content Manager Resource Manager for z/OS*, GI10-8531.

In these program directories, there are prerequisite PTFs for other products such as DB2. Also, there are several PTFs for Content Manager V8.2. Contact the Support Center to get information on all available PTFs that should be applied.

In Appendix B of each of these documents, there is information about the newly provided CLISTs/edit macros which help you in customizing your JCL. To work with this macro during your edit session, you must place it in a library included in the SYSPROC concatenation of your system. To find out which libraries are in the SYSPROC concatenation, enter the ISRDDN command in the ISPF command shell (option 6).

Also in Appendix B of the *Program Directories*, there is a list with a description of all the replacement tokens used in the JCL. This information is very useful. Unfortunately it is not placed in the manual, *Planing and Installing Your Content Management System for z/OS*, GC18-7698. If you have to do the customization and do not have the *Program Directory* available, ask your system programmer.
To make the customization process of the JCL even easier, you can create one single edit macro to change all the replacement tokens in the JCL; there is no difference between Library Server and Resource Manager. For our test, we use one macro which can be found in Appendix A, “Installation and configuration overview (for V8.2 only)” on page 601.

**Notes:** Please keep the following considerations in mind:

- Have all the necessary manuals available. Assuming you have OAM already installed and in operation, the minimum manuals you need are:
  - *Program Directory for IBM DB2 Content Manager Library Server for z/OS*, GI10-8530
  - *Program Directory for IBM DB2 Content Manager Resource Manager for z/OS*, GI10-8531
  - *Planning and Installing Your Content Management System for z/OS*, GC18-7698
  - *IBM DB2 Content Manager for Multiplatforms System Administration Guide*, SZ27-1335

- As most of the work during the implementation refers to DB2, it is also important to work closely with the DB2 administrator and to have a user ID with the required authorities assigned.

- Resource Manager is a program which runs under the HTTP server. Skill in this area is also required.

**Important:** If you already have a Content Manager Version 2.3 system running and you are planning for a migration, read the manual *Migrating to Content Manager Version 8 for z/OS*, GC18-7699, before starting with the implementation.

### A.4.2 Performing the implementation

After the installation using SMP/E, there will be several SICM* libraries on your system. The following libraries (libraries with 1 at the end refer to the Library Server, with a 2 to the Resource Manager) are important for further implementation:

- **SICMCPY1**: Copybooks for installation
- **SICMDBR1**: DBRMs for the Library Server
- **SICMINS1**: Installation and implementation JCL
- **SICMLMD1**: Load modules
In addition to these libraries, you must define some additional user libraries. These libraries are:

- **ICMULOAD**: User load libraries containing access modules
- **ICMUDBRM**: User DBRM library
- **PROCLIB**: Procedure library containing user table access program and SQL procedures for creating the user tables
- **ICMUPGM**: User program library containing source of access modules
- **ICMUSQL**: User library containing the DDLs for the item types
- **ICMUOBJ1**: User object library needed during compile and linkedit

**Note:** Library ICMUOBJ, according to *Planning and Installing Your Content Management System for z/OS*, GC18-7698, is predefined; nevertheless, we did not find any reference to this library and therefore did not define it. During the tests we performed, this had no impact on the system.

For allocation details such as DCB parameters, refer to *Planning and Installing Your Content Management System for z/OS*, GC18-7698.
Tip: With the supplied JCL, the names of the libraries ICMULOAD, ICMUDBRM, and PROCLIB must have exactly these LLQ. If you want to give them any other LLQ, only small changes must be made in the following jobs. For ICMULOAD and ICMUDBRM, change job ICMMLSPP as follows:

Change the PROC-statement of the JCL:

- remove
  ```
  //         APPLHLQ='?APPLHLQ?',
  ```
- add
  ```
  //         ICMUDBRM='?ICMUDBRM?',
  //         ICMULOAD='?ICMULOAD?',
  ```

Then, change the following three statements:

- Change `//DBRMLIB DD DSN=&APPLHLQ..ICMUDBRM(&MEMBER),DISP=SHR` to `//DBRMLIB DD DSN=&ICMUDBRM.(&MEMBER),DISP=SHR`
- Change `//SYSLMOD DD DSN=&APPLHLQ..ICMULOAD(&MEMBER),DISP=SHR` to `//SYSLMOD DD DSN=&ICMULOAD.(&MEMBER),DISP=SHR`
- Change `//DBRMLIB DD DSN=&APPLHLQ..ICMUDBRM,DISP=SHR` to `//DBRMLIB DD DSN=&ICMUDBRM.,DISP=SHR`

For PROCLIB, change the JCLLIB statement in jobs ICMMLSP1, ICMMLSP2 and ICMMLSP3 as follows:

```
//LIB JCLLIB ORDER='?ICMPROC?'
```

After making these changes, you can select any name for all of the 5 (6) libraries which have to be allocated.

For our test, we customize the jobs found in the SICMINS* libraries using our edit macro and change all the replacement tokens in the JCL.

**Library Server**

Run the jobs according to the instructions given in Step 1 to Step 4 of the *Planning and Installing Your Content Management System for z/OS, GC18-7698*.

Note: If you are planning for a migration, keep in mind that the user IDs already defined in the running system must not be defined in the system you are migrating to. For example, if you have ICMADMIN already defined in your V7 system, do not use this ID again for the V8 installation.
As in the previous versions of Content Manager, you have to decide how the process of creating item types (formerly index classes) should be done. There are still three different ways to prepare the user table access program:

- Execute the user table access program and the corresponding SQL.
- Copy the JCL for the access program preparation and the SQL in PDS members and submit them later.
- Execute and copy the JCL (combination of the two proceeding).

For our test, we use the option to copy and not to execute automatically. This gives us more control of what is going on. If you choose this option, however, you have to keep it in mind, especially during the testing period. It may be a good idea for your test environment to change this, after you have carefully tested the mechanism, back to the automatic creation. Depending on which way you choose, you have to customize your WorkLoadManager (WLM) procedure accordingly. Also, the corresponding JCL procedures have to be updated.

As the Library Server in Version 8 is a set of DB2 stored procedures and functions, it is important to have the WLM environment set up correct. This task should be performed by the DB2 administration people.

Normally, the setup of the WLM is the same as usual for the DB2 administrator and it should done very quickly. We customized the procedure for our own use and it works fine.

For our test, we decide not to use the logon user exit routine for the Library Server for z/OS, not to use the batch import/synch utility, not to use the two access control user exit routines, and also not to use the monitor program.

**Note:** For the z/OS environment, there is one table which allows you to define additional clauses to the CREATE TABLE and CREATE INDEX statements generated by the Library Server. This table is the ICMST390Control table. With this table, you have the flexibility to add any valid DB2 clause to the generated SQL.

The table has only two columns, Option Key and Option Text. In Option Key, you can define TABLE or INDEX. In the Option Text, you can define what clause should be added to the SQL. You can also use any valid Variable which will be replaced by the stored procedure, for example +COMPID+.

The processing flow of the stored procedure is:

reading table --> creating SQL --> replacing variables
**Resource Manager**

We continue with the implementation of the Resource Manager.

With Version 8, the Resource Manager is a single dynamically loaded module running under the HTTP server for z/OS. In this matter, it is very much similar to a CGI program. Most of the work for the Resource Manager implementation has to do with the setup of the HTTP server. With the provided sample code, this task should be easier.

**Attention:** In the *Planing and Installing Your Content Management System for z/OS*, GC18-7698 on page 42, there is a description of the SY SIN parameters OBJTOKENS and TRACELEVEL.

This is wrong if you have applied PTF UQ79741 for the Library Server and PTF UQ79742 for the Resource Manager. These two parameters are removed from the SY SIN file and can now be found as columns of the ICMRMCONTROL table.

Beside the one program which runs under the HTTP server, Resource Manager also has a database with five tables. These tables are used for the following purposes:

- Tracking table for rollback or synchronization with the Library Server
- Encryption key for the Library Server
- Asynchronous replication requests to other Resource Managers
- Control table for the use of exit programs and trace level (the control table also contains the prefetch enabled column)
- Prefetch table

JCL is provided in SICMINS2 to create the database and tables. Also, for the bind and the grant, JCL is available. We customize this JCL with the macro and everything works fine.

**Tip:** The replacement tokens ?ICMMOPLN? and ?DB2PLAN? refer both to the Resource Manager plan name and must be set to the same identifier.

There are several Resource Manager pre-exits and post-exits for store, prefetch, replace, retrieve, delete, query, and change SMS. For example, these exits can provide a capability to override the collection name or SMS construct. For the initial setup of the Resource Manager, for our test, we decide not to use any of the exit programs. This can be done in a later step. Information about the Resource Manager exit programs can be found on page 42 of the manual, *Planing and Installing Your Content Management System for z/OS*, GC18-7698.
The Library Server and the Resource Manager run on the same system; therefore, there is no need to define a remote Library Server.

For our test, we start the HTTP server and it comes up.

**Attention:** The installation verification description on page 47 of the *Planning and Installing Your Content Management System for z/OS*, GC18-7698 is a bit too brief.

Be aware that the first point of that list, “Install and configure your client workstation (either Client for Windows or eClient)”, in reality requires much more work than it seems. This includes all the tasks described in “Connecting the System Administration Client to z/OS”, starting on page 51 of the *Planning and Installing Your Content Management System for z/OS*, GC18-7698.

**Setting up the workstation**
The next step is to set up the workstation. First, we install the System Administration Client software. This process is almost completely automatic. For our test, we accept all the default paths.

For our test, DB2 is already installed; we do not have to perform any DB2 installation. If your workstation does not have DB2, perform the installation now. For the System Administration Client and the Windows client, you need the runtime environment and DB2 Connect; there is no need to install the entire DB2 product.

Next you need to catalog the database. This is easy to do with the DB2 Configuration Assistant. See page 52 of the *Planning and Installing Your Content Management System for z/OS*, GC18-7698 for details.

Sometimes the same name is used for different things. Table A-4 helps you in translating the Windows identifier to what it is on the host.

<table>
<thead>
<tr>
<th>Windows identifier</th>
<th>Host equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host name</td>
<td>IP address of your host system (also the symbolic name can be used); can be looked up in the subsystem job xxxxMSTR, search for DOMAIN</td>
</tr>
<tr>
<td>Database name</td>
<td>LOCATION of the DB2 subsystem; can be looked up in the subsystem job xxxxMSTR, search for LOCATION NOT the Library Server database name</td>
</tr>
</tbody>
</table>
Appendix A. Installation and configuration overview (for V8.2 only)

At the end of the catalog process, a test of the connection is performed. If this test works, you can access the server database from your client. That is a major step to accomplish.

Next, you have to configure the server for the Content Manager clients.

<table>
<thead>
<tr>
<th>Windows identifier</th>
<th>Host equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>Port where the DB2 subsystem is listening to; can be looked up in the subsystem job xxxxMSTR, search for TCPPORT</td>
</tr>
<tr>
<td>Database alias</td>
<td>Your free choice</td>
</tr>
<tr>
<td>System name</td>
<td>Your free choice</td>
</tr>
</tbody>
</table>

**Note:** There is no information required for the following fields:
- Service name
- Remote instance name

At the end of the catalog process, a test of the connection is performed. If this test works, you can access the server database from your client. That is a major step to accomplish.

**Important:** We do not recommend to perform this setup as it is described in Step 2 on page 52 of the *Planning and Installing Your Content Management System for z/OS*, GC18-7698. Editing the text of an INI-file with cut and paste and modifying some values requires a very good knowledge of the file and what it does. Also, when first installing your system, there will not be any server definition in that file.

We strongly recommend using the Server Configuration Utility!

To configure the server using the Content Manager Server Configuration Utility, click **Start** → **Programs** → **IBM Content Manager for Multiplatforms V8.2** → **Server Configuration Utility**. An Input Screen (see Figure A-4) appears, and you can enter the information for your system.

**Tip:** There is one reason to edit the \icmroot\ICMgmt\CMBICMSRVS.INI file, and this is because the available servers in the pull-down menu after starting the System Administration Client are sorted in the order they are listed in this file. If you want to change the order, you have to edit this file. Do this with care and always move all the information regarding one server together.
Click **OK**. After this, it is possible to log on with the System Administration Client to the z/OS Library Server. At this time, it is a good idea to set the password for the user ID of the Resource Manager administrator. Just click **Resource Manager**, do a right-click on the name of your Resource Manager in the right side window, then select **Explore** and enter the valid password for the user ID. During the implementation, the password which was inserted in the DB2 table is a non-valid password, and you have to type in the correct password in this window. Click **OK** or **Apply** and **Cancel**.

**Note:** Do not start defining item types, as the system item types are not yet defined.
Open a command window on your workstation and enter these commands:

- Install drive `\icmroot\cmbHcmenv81`
- Install drive `\icmroot\CM82\config\JAVA TRebuildCompTypeICM databasealias userid password schema errorfile`
- Install drive `\icmroot\CM82\config\JAVA ICMDefineSystemItemTypes databasealias userid password schema errorfile`

**Attention**: The `CMBICENV81.bat` file has to be customized before running it.

Example 21-5 shows the result of these commands.

**Example 21-5  Output of TRebuildCompTypeICM and ICMDefineSystemItemTypes**

- Number of views found: 15
- Generating access module for view with ID: 200
- Generating access module for view with ID: 201
- Generating access module for view with ID: 202
- Generating access module for view with ID: 203
- Generating access module for view with ID: 204
- Generating access module for view with ID: 205
- Generating access module for view with ID: 206
- Generating access module for view with ID: 207
- Generating access module for view with ID: 208
- Generating access module for view with ID: 300
- Generating access module for view with ID: 301
- Generating access module for view with ID: 302
- Generating access module for view with ID: 303
- Generating access module for view with ID: 304
- Generating access module for view with ID: 400
- All access modules rebuilt

**Note**: The second command, `ICMDefineSystemItemTypes`, does not give you any message on the screen; but, if you look in the log of the host machine, you find the following message (of course, with your IP address and DB2 name):

```
DSNL511I  -DB2C DSNLIENO TCP/IP CONVERSATION FAILED 098 TO LOCATION 9.30.40.236
IPADDR=9.30.40.236 PORT=2279 SOCKET=RECV RETURN CODE=1121 REASON CODE=00000000
```

Do not be confused by this message. Just look at the job logs to see whether the jobs ran successfully or not.
Depending on how you have set up your procedure for creating the tables and the table access programs, you probably have to submit the jobs generated and stored in the PDS. For our test, we submit the jobs and they run.

**Note:** The SQL jobs must run first and the user ID under which the create jobs run must have the privilege to create tables.

The user ID for the generation of the access program must have DB2 bindadd authority.

### A.4.3 Installation verification

To verify installation, log on using the Windows Client and store two documents in the NOINDEX item type. Do a search and see if you can get both documents in the result list. If you do, click on one of them, you should get the error message shown in Figure A-5.

![Figure A-5  Unable to open Image](image)

To get the client working, it is necessary to update the ICMClient.INI file. On a Windows 2000 system, this file is placed in the C:\WINNT directory.

Update the file to be similar to the following coding:

```ini
[Options]
DisableURLCode=yes
DisableEditIndexSaveOption=no
LoadEditIndexKeyfieldsFromScratch=yes

[Scan]
PagesInFirstFile=5
PagesInRemainingFiles=15
```

Save the file, start the client again, and now it should work.
A.4.4 Summary

The implementation of the system can be done in a straightforward manner.

Because there is no more need for CICS and all the definitions required there, the implementation can be performed in a few days; nevertheless, high DB2 authority is needed during the process. Support of the HTTP personnel is also very important.

For our test, the installation of Version 8 was quite smooth.
Migration on z/OS  
(for V8.2 only)

In this chapter, we describe how to migrate to Content Manager for z/OS V8.2, from planning to realization. Shipped with the product, there are several batch jobs which help you in the migration process. These jobs assist you in either migrating from Content Manager for z/OS V2.3 with PTF UQ50981 applied, which is mostly called Content Manager for z/OS V7, or from Content Manager ImagePlus for OS/390.

We show what to do when planning the migration, including a checklist of the information you must gather before starting with the migration process. In addition, we go through the migration steps and explain the important areas you must take care of during the process.

**Important:** This chapter covers the migration from previous version of Content Manager to Content Manager V8.2 on z/OS only. It is kept here for reference purposes.

To get the latest migration updates for Content Manager V8.3 for z/OS, refer to the product manuals and the following redbook:

*DB2 Content Manager for z/OS: Implementation, Installation, and Migration, SG24-6476*
B.1 Migrating from Content Manager for OS/390 V2.3

Coming with Content Manager for z/OS V8.2, there are two tool sets shipped that help with the migration. One tool set is for the migration of a Content Manager for OS/390 V2.3 system. The other toolset is for the migration of a Content Manager ImagePlus for 390 V2.2 and higher system.

We cover the Content Manager for OS/390 V2.3 migration in this section. We cover the ImagePlus migration in B.2, “Migrating from CM ImagePlus for OS/390” on page 656.

**Note:** The migration process migrates only the meta data and not the objects. Once the migration has started, the Content Manager for OS/390 V2.3 system cannot be used.

Apply all available PTFs to your system. Right after the Content Manager V8.2 product GA, several very helpful tools for the migration have been added. We use GA code with all PTFs applied.

This chapter of the redbook should be used in conjunction with the following IBM manual:

- *Migrating to Content Manager Version 8 for z/OS*, GC18-7699

We perform the migration tasks according to the step-by-step instructions given in this manual. There are eight steps involved:

- “Step 1: Verify existing data” on page 650
- “Step 2: Upgrade Content Manager” on page 651
- “Step 3: Upgrade DB2” on page 651
- “Step 4: Verify SMP/E post-installation job ICMMLSLD” on page 651
- “Step 5: Install Content Manager for z/OS Version 8.2” on page 651
- “Step 6: Generate system item and component types” on page 652
- “Step 7: Define any additional Resource Managers” on page 652
- “Step 8: Shut down Library Server” on page 652

Please refer to the manual for detailed descriptions. We document how we perform the steps as follows.

**Step 1: Verify existing data**

This step is very important for importing data to the new environment later. If your migration data is not correct, you may encounter many problems later in the migration process.
In our scenario, we checked to make sure:

- That there were no extra rows in tables AVT00004 and AVT00005.
- That there was no mismatch between the DB2 and the Content Manager database definitions.

**Tip:** The SQLs for the queries used to verify the data can be found in member ICMMMI72 of library SICMMIN1.

**Step 2: Upgrade Content Manager**

One of the prerequisites for the migration is that you have Content Manager Version 7 on your system. Content Manager Version 7 is Content Manager Version 2 Release 3 with PTF UQ50981 applied.

In our scenario, we already have this system level installed.

**Step 3: Upgrade DB2**

For Content Manager V8.2 system, you must have DB2 Version 7.1 or later. If not, you need to upgrade to at least the minimum level.

In our scenario, we already have this version installed.

**Step 4: Verify SMP/E post-installation job ICMMLSLD**

At this point, it is very important to go through the migration manual before you perform the actual installation of the Content Manager Version 8.2 system. We have already created some definitions in our Version 8 environment for testing purposes, and at this point, we need to delete all these definitions.

Because the default Object Server for Content Manager Version 7 system is required to become the default Resource Manager for Content Manager Version 8.2 system, we recommend not to save any objects in the new Content Manager Version 8.2 system at this point.

In our scenario, we deleted our installation and set up a new Content Manager V8.2 system from scratch.

**Note:** It is not sufficient just to drop the database. The stored procedures and functions must explicitly be dropped as well. Job ICMMLSDR in SICMINS1 and job ICMMRMRMDR in SICMINS2 are prepared for this.

**Step 5: Install Content Manager for z/OS Version 8.2**

We have all the jobs ready from the prior installation. So we dropped the functions, procedures, and databases, and installed everything again.
**Step 6: Generate system item and component types**

We entered the following commands:

```
Install drive \icmroot\cmbicenv81
Install drive \icmroot\CM82\config\JAVA TRebuildCompTypeICM databasealias userid password schema errorfile
Install drive \icmroot\CM82\config\JAVA ICMDefineSystemItemTypes databasealias userid password schema errorfile
```

**Step 7: Define any additional Resource Managers**

We have only one Object Server to migrate. We did not have to perform additional tasks here.

**Step 8: Shut down Library Server**

The Library Server was already shut down.

### B.1.1 Preparing the migration jobs

The next task is the customization of the migration JCL. There are six steps involved:

- “Step 1: Map Object Servers with Resource Managers” on page 652
- “Step 2: Customize migration jobs” on page 652
- “Step 3: Create migration tables” on page 652
- “Step 4: Bind the migration package” on page 652
- “Step 5: Migrate system definition and create user tables” on page 653
- “Step 6: Create directory for export to workstation Object Servers” on page 653

Please refer to the manual for detailed descriptions. We document how we perform the steps as follows.

**Step 1: Map Object Servers with Resource Managers**

We use the sample SQL from the migration guide.

**Step 2: Customize migration jobs**

We use our edit macro from the customizing of the installation jobs. There are only a few variables to add and then the macro works fine.

**Step 3: Create migration tables**

The job completes successfully. Depending on your system environment, you must customize the table space names and sizes.

**Step 4: Bind the migration package**

This job runs smoothly without any special steps.
Step 5: Migrate system definition and create user tables

If this job does not end successfully, the entire Content Manager database has to be dropped and rebuild. In our scenario, fortunately, we do not have to do that.

Tip: If this job does not end normally, usually you get some error codes at the end of the output. Depending on the error, take appropriate actions.

Step 6: Create directory for export to workstation Object Servers

Even though we do not have this kind of servers, we do as on requested.

B.1.2 Running the migration jobs

There are seven steps involved here. Please refer to the manual for detailed descriptions. We document how we perform the steps as follows.

Important: Before continuing with the migration, job ICMMMI71 to rebind the packages and job ICMMCACL from library SICMINS1 have to run. This is documented in the ICMMMI72 JCL.

Step 1 to 6: Migrate the data

Follow the migration documentation to perform these steps. We did not get any errors.

Attention: With PTF UQ79741, the migration procedure has changed. This is documented in the comment section of the PTF; but it can easily be overlooked. The main change is an additional job, ICMMMI77. This job has to run at the end of the migration.

Unfortunately, after install of PTF UQ79741, the JCL of ICMMMI77 is placed in SICMMDB1. This should be fixed with PTF UQ81734. It is very important to apply all available PTFs to the system.

If you get errors, refer to the migration documentation about what the jobs are doing and what to do when errors occur. In addition, we provide a special section, B.1.4, “Migration hints and tips” on page 654, to assist you for your migration process. The section also give you additional information regarding performance.

Step 7: Export data to workstation Object Servers

We do not have any non-z/OS Object Server defined to the Content Manager V7. system. We skip this step in our scenario.
B.1.3 Post migration validation

Prior to migration, you should have a set of testing suites that you used to validate your current system is operating correctly. Make sure that you perform the same set of testing suites on your new system after the migration to validate that everything is migrated successfully and the system is operating as expected.

In our scenario, we test our new migrated system by performing search and retrieve functions on a known set of documents. This works as we expected. We also test storing new documents, defining new item types, and searching and retrieving against the new documents. They all work very well. Of course, the test we run is not an extensive test suite; but with our limited time frame, this simple set of tests is sufficient for us, as the post migration validation.

B.1.4 Migration hints and tips

Based on our experience, in this section, we include some migration hints and tips that we find are important that may assist you for your migration. They include the following references:

- “Verify your data” on page 654
- “Check the prerequisites” on page 654
- “Define additional Resource Managers” on page 654
- “Define non-used user ID” on page 655
- “Run the jobs in parallel” on page 655

Verify your data

It is very important that you verify the existing Version 7 data before you start with the installation. Although this is mentioned in “Step 1: Verify existing data” on page 650, we want to re-emphasize the importance of this step. Refer to Chapter 3 of the Migrating to Content Manager Version 8 for z/OS, GC18-7699 for more details.

Check the prerequisites

Make sure that you are at the required PTF level with your Content Manager Version 2.3 system. The required level is PTF UQ50981 applied.

Define additional Resource Managers

Make sure you define the additional Resource Managers as needed. Again, refer to Migrating to Content Manager Version 8 for z/OS, GC18-7699 for more details.
Define non-used user ID
Define one user ID in your Content Manager Version 8 system before you start the migration. Make sure this user ID has SYSADMIN privilege and userprivsetcode = 1. Also, make sure that it is not a user ID already existing in your Content Manager Version 7 system.

Run the jobs in parallel
After the migration jobs ICMMMI72 and ICMMMI73 have completed, the rest of the jobs can run in parallel. With this, you can reduce the elapsed time.

Note: If you do not run the jobs in parallel, it takes a longer time to complete the rest of the jobs. In our test scenario where we ran the jobs in parallel, the migration of meta data (for 1.5 million objects) took about 1.5 hours. Actually, this number is from the test scenario in our environment. The time is probably different in your environment. Do not use this number for estimating the hours that you will need for your migration.

Figure B-1 illustrates where and what job(s) can run parallel.
Notice the following considerations:

1. All the ICMUJOBS submitted by the Library Server during ICMMMI72 must be completed first. If your system setup does not automatically run these jobs, you have to manually submit them.

2. If each Content Manager Version 8.2 user table is defined in its own table space, multiple ICMMMI7A jobs can be submitted to migrate multiple AVT tables in parallel.

3. If you have set up the system to use a separate table space for each user table, these table spaces must be defined before running the ICMUJOBS.

**Tip:** If there is an error during the migration, the error code may have the following structure:

   xxxxx, yyyy, other information text

Where:

   yyyy is a Library Server code.
   xxxxx is a migration error code.

Look up the yyyy error codes in the messages and codes manual for the Library Server.

For the xxxxx error codes, they are currently not listed in any publication. They can be found in a file named migration.properties (there is a different one for every language). This file is included in the icmndfsy.jar file which is in the migration directory (\migrate\db2) of the Content Manager server CD.

To view the contents of the JAR file, you can either enter:

   jar tf icmndfsy.jar

Or, copy the file to icmndfsy.zip and open it with a data compression utility such as WINZIP or PKZIP.

### B.2 Migrating from CM ImagePlus for OS/390

In this section, we provide information on migrating from IBM Content Manager ImagePlus for OS/390 to IBM Content Manager for z/OS Version 8.2. During the writing of this redbook, due to time limitations, we do not actually perform this scenario; however, we provide hints and tips collected from other customer projects, where this migration has been performed. We hope this information will help you to perform a smooth migration.
For a detailed description of the migration process, including the prerequisites necessary to do the migration, refer to *Migrating to Content Manager Version 8 for z/OS*, GC18-7699. In chapter 2 of this manual, you can find all the information needed for planning and executing the migration.

### B.2.1 Hints and tips

In the following sections we provide several hints and tips to help you in your migration:

- “Library - Sample JCL location” on page 657
- “Multiple runs for migration” on page 657
- “Sample values - Finding the next available number” on page 657
- “Overlays - Not migrated” on page 658
- “Annotations - Migrated” on page 658
- “Object accessibility - Accessible from both systems” on page 658
- “Object status (OBJSTAT) setting in ImagePlus” on page 658
- “Migration to Multiplatforms” on page 659

**Library - Sample JCL location**

The sample JCL for the migration process, referred to in *Migrating to Content Manager Version 8 for z/OS*, GC18-7699, is located in the SICMMIN1 library.

**Multiple runs for migration**

The way the migration from Content Manager ImagePlus for OS/390 to Content Manager for z/OS V8.2 is performed is totally different from the V7 to V8.2 migration for Content Manager for z/OS. The way the ImagePlus migration is performed is similar to unloading data and then inserting data; therefore, you do not have to migrate all of your ImagePlus data in one single run, and you have the option of running in phases.

This is also described in *Migrating to Content Manager Version 8 for z/OS*, GC18-7699. Refer to this manual for more details.

**Sample values - Finding the next available number**

The sample values you find in the migration guide, such as the item type ID, will work, if you have an empty Content Manager for z/OS system. This means that you have installed your Content Manager for z/OS, and have run the TRebuildItemTypeICM and the ICMDefineSystemItemType program.

If you have already defined additional item types, you must find out the next number you can use for the item types created during the migration. You can do this by performing a query against the ICMSTNLSKeywords table of your Content Manager for z/OS and selecting the maximum of the column keywordcode for the corresponding keyword class.
For example: “What is the number for the next item type?” assumes you are using ENU as the language code. You should use the following query:

```sql
SELECT max(KEYWORDCODE) FROM ICMSTNLSKeywords
WHERE KEYWORDCLASS = 2 and LANGUAGECODE = 'ENU';
```

The result is the last code already used by the system. You have to add 1 to this number and the new number becomes the next available number.

**Overlays - Not migrated**

If you have Overlays in your ImagePlus system, these overlays are not migrated.

**Annotations - Migrated**

Annotations are migrated.

**Object accessibility - Accessible from both systems**

Objects migrated to the new Content Manager for z/OS system are still accessible from the existing Content Manager ImagePlus for OS/390 system. Together with the partial migration, this gives you more flexibility in planning your migration.

**Important:** An object can remain active in the old ImagePlus system after being migrated to new CM V8.2 system and it may be accessed via IPFAF and IWPM, or the eClient; it may also be included in a federated eClient search list.

Remember, in this situation, the changes made to the object or its index information in the old ImagePlus system are not visible in the new CM V8.2 system, and the changes made in the new CM V8.2 system are not visible in the old ImagePlus system.

**Object status (OBJSTAT) setting in ImagePlus**

The ImagePlus object table, EYPTOBJTxx, has an OBJSTAT field that IPFAF either set to “0” or “1”

Where:

“0” means that the object is fully active and available for display, update, and workflow.

“1” means that the object is “logically” deleted and is not included when request a folder contents list.

We recommend that once objects are migrated and no longer needed in IPFAF, then set this field to “2”.

---

658   Content Manager Implementation and Migration Cookbook
A “2” is treated the same as a “1” by IPFAF, but it gives the user the capability to recover the migrated objects entries if that level of determination is necessary.

**Migration to Multiplatforms**

A frequently asked question is, if ImagePlus data can be migrated to CM on Multiplatforms. The simple answer is, yes; it should work. The flow has not been tested, but careful consideration needs to be taken if this path is pursued. The JCL for the migration utilities is organized in three major groups:

- ICMMMIFx jobs unload the applicable ImagePlus tables.
- ICMMMPx jobs process the unloaded data into a format acceptable to CM.
- ICMMMICx jobs create the required CM tables, load attribute data, and load the output from the ICMMMPx jobs.

The ICMMMIFx and ICMMMPx jobs can still be used. After running these jobs, the process varies. Using the Content Manager System Administration Client, the proper attributes, ItemTypes and Components, need to be defined. The JCL and control data should be used as a guide to this process. Notice that the definitions need to match exactly for the DB2 load to work.
Appendix C. Replacement token for z/OS installation JCL CLIST/edit macro

This appendix contains the edit macro we use during the customization. We put all the variables together in one file. This makes it easier for us to work with the macro.

ISREDIT MACRO
ISREDIT UP M
ISREDIT CHANGE (XXXX) () ALL
ISREDIT CHANGE ?ACLPLAN? IGFCACL ALL
ISREDIT CHANGE ?BPASY? BP32K ALL
ISREDIT CHANGE ?BPCTL? BP32K ALL
ISREDIT CHANGE ?BPNAME? BPO ALL
ISREDIT CHANGE ?BPPFT? BPO ALL
ISREDIT CHANGE ?BPS4? BPO ALL
ISREDIT CHANGE ?BPTRK? BP32K ALL
ISREDIT CHANGE ?BPV4? BPO ALL
ISREDIT CHANGE ?BP32? BP32K ALL
ISREDIT CHANGE ?CCHLQ? CBC ALL
ISREDIT CHANGE ?CREATOR? GFICHTI ALL
ISREDIT CHANGE ?CEE? CEE ALL
ISREDIT CHANGE ?DATABASE? IGFLS390 ALL
ISREDIT CHANGE ?DB2CONNECTID? IGFCNCT ALL
ISREDIT CHANGE ?DB2PKGCOLLID? IGFCLS82 ALL
ISREDIT CHANGE ?DB2PLAN? IGFMOPLN ALL
ISREDIT CHANGE ?DB2RUN? DB2C.RUNLIB.LOAD ALL
ISREDIT CHANGE ?DB2SSID? DB2C ALL
ISREDIT CHANGE ?DSN? DB2C ALL
ISREDIT CHANGE SDSNLOAD DSNLOAD ALL
ISREDIT CHANGE ?DSNRUN? DB2C.RUNLIB.LOAD ALL
ISREDIT CHANGE ?DSNTIAD? DSNTIAD ALL
ISREDIT CHANGE ?GROUP00? GROUP00 ALL
ISREDIT CHANGE ?GROUP01? GROUP01 ALL
ISREDIT CHANGE ?GROUP02? GROUP02 ALL
ISREDIT CHANGE ?GROUP03? GROUP03 ALL
ISREDIT CHANGE ?HLQ?.I GFICHTI.CM82.S ALL
ISREDIT CHANGE ?HLQ? GFICHTI.CM82 ALL
ISREDIT CHANGE ?ICM? GFICHTI.CM82 ALL
ISREDIT CHANGE ?ICMADMIN? GFICHTI ALL
ISREDIT CHANGE ?ICMLHLQ? GFICHTI.CM82 ALL
ISREDIT CHANGE ?ICMINS1? GFICHTI.CM82.UPROCLIB ALL
ISREDIT CHANGE ?ICMLIBRARYSERVERID? 10 ALL
ISREDIT CHANGE ?ICMLLOG? GFICHTI.CM82.ICMLOG ALL
ISREDIT CHANGE ?ICMMLSP? ICMMLSP2 ALL
ISREDIT CHANGE ?ICMMLSQ? ICMMLSQ2 ALL
ISREDIT CHANGE ?ICMMOPLN? IGFMOPLN ALL
ISREDIT CHANGE ?ICMMOSAP? IGFMOSAP ALL
ISREDIT CHANGE ?ICMMOSAR? IGFMOSAR ALL
ISREDIT CHANGE ?ICMMOSDI? IGFMOSDI ALL
ISREDIT CHANGE ?ICMMRDBM? IGFRM390 ALL
ISREDIT CHANGE ?ICMPROC? GFICHTI.CM82.UPROCLIB ALL
ISREDIT CHANGE ?ICMULOAD? GFICHTI.CM82.USERLOAD ALL
ISREDIT CHANGE ?ICMUDBRM? GFICHTI.CM82.USERDBRM ALL
ISREDIT CHANGE ?ICMUPGM? GFICHTI.CM82.USERPGM ALL
ISREDIT CHANGE ?ICMUSQL? GFICHTI.CM82.USERSQL ALL
ISREDIT CHANGE ?LEHLQ? CEE ALL
ISREDIT CHANGE ?LSDBLOC? NQADB2C ALL
ISREDIT CHANGE ?LSDBNAME? IGFLS390 ALL
ISREDIT CHANGE ?LSSSCHEMA? GFICHTI ALL
ISREDIT CHANGE ?MSGCLASS? H ALL
ISREDIT CHANGE MSGCLASS=H 'MSGCLASS=H,NOTIFY=&SYSUID' ALL
ISREDIT CHANGE ?NUMTCB? 20 ALL
ISREDIT CHANGE ?OAMDBRM? SYS1.CBRDBRM ALL
ISREDIT CHANGE ?OUTC? H ALL
With these replacements, we customized all our jobs.
ACL user exits UDF declarations

This appendix specifies the UDF declarations for ICMACLPrivExit and ICMGenPrivExit ACL user exits.

**ICMACLPrivExit**
The UDF declaration for the ACL privilege exit is as follows:

```c
void SQL_API_FN ICMACLPrivExit ( 
    char *pszUserID, 
    char *pszApplicationID, 
    char *APIAction, 
    char *pszProcedureName, 
    char *InfoType, 
    char *ItemID, 
    char *ViewID, 
    long *plGenPrivCode, 
    short *pszUserID_ind, 
    short *pszApplicationID_ind, 
    short *APIAction_ind, 
    short *pszProcedureName_ind, 
    short *InfoType_ind, 
    short *ItemID_ind, 
    short *ViewID_ind, 
    short *plGenPrivCode_ind, 
);```

© Copyright IBM Corp. 2004, 2006. All rights reserved.
char *sqlstate, /*SQL STATE */
char *funcname, /*function name */
char *specname, /*specific function name */
char *mesgtext /*message text insert */
)

Input parameters:

pszUserID User ID (32 characters)
pszApplicationID Application ID (32667 characters - reserved)
pszHostname Host name of client (128 characters; API will pass in maximum 151 characters)
APIAction API action:
  0 Retrieve
  2 Update
  3 Delete
  4 Not available
pszProcedureName Library Server stored procedure name (20 characters)
InfoType
  1 Item related
  2 View related
  3 Item type or item type view related
ItemID Item ID (26 characters)
ViewID Item type ID, item type view ID, or view ID (may not be available to retrieve spanning multiple views)
pICGenPrivCode General Privilege required for the action

Output parameters:

pIRC Pointer to return code

Return codes:

  0 RC_OK, validation o.k. ICMACLPrivExit grants the ACL to query
  Any other return code declines access to the ACL query

ICMGenPrivExit

The following listing is the GenPrivExit function. The listed parameters provide information available for use in your security access logic:

extern long ICMGenPrivExit(
  char *pszUserID,
  char *pszApplicationID,
  long *pICGenPrivCode,
  char *pszProcedureName,
  long *pICMDecision,
  long *pICMDecReason,
  long *pIRC)
Input parameters:

- **pszUserID** User ID (32 characters)
- **pszApplicationID** Application ID (32667 characters - reserved)
- **pszHostname** Host name of client (128 characters; API will pass in maximum 151 characters)
- **plGenPrivCode** General privilege required for the action
- **pszProcedureName** Library Server stored procedure name (20 characters)
- **plICMDecision** Library Server decision
  - 0 Grant access
    - Look up any other return code
- **plICMDecReason** Extended return code if available

Output parameters:

- **plIRC** Pointer to return code

Return codes:

- 0 **RC_OK**, validation o.k. ICMGenPrivExit grants general privilege for the API action.

To enable ACL user exit checking, go to the System Administration Client and open the Library Server Configuration dialog box. Under the Features tab, check the **Enable ACL User Exit** box.

**Important:**

- Do not turn on the flag unless the exits have been compiled and linked.
- In ICMACLPrivExit() the ViewID parameter may not be available in queries involving multiple active views. In these situations, a negative number is instead sent to the user exit.
API migration tables for Content Manager

Applications developed for Version 7 cannot be used with the ICM Datastore. They must be rewritten with the new object-oriented APIs to exploit the new features of Content Manager Version 8.

In this appendix, we provide tables to help you in porting your application to the new APIs. These tables show:

- Which APIs are the same as, or similar to, those used in the prior version
- What new functionalities have been introduced
- What functions are no longer supported
- References to sections in the Workstation Application Programming Guide, where you can find more details about the particular APIs (or functions)

The column labeled “DL Connector” shows the method calls used in older Content Manager versions. The column labeled “ICM Connector” shows the method calls used in the new Version 8.
Table E-1  API migration table for Content Servers

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log on and log off</td>
<td>SimLibLogon()</td>
<td>DKDatastoreDL.connect()</td>
<td>DKDatastoreICM.connect()</td>
</tr>
<tr>
<td></td>
<td>SimLibLogoff()</td>
<td>DKDatastoreDL.disconnect()</td>
<td>DKDatastoreICM.disconnect()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Connecting to content servers</td>
</tr>
<tr>
<td>List Data sources</td>
<td>Ip2ListServers()</td>
<td>DKDatastoreDL.listDataSources()</td>
<td>DKDatastoreICM.listDataSources()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Connecting to content servers</td>
</tr>
<tr>
<td>List Object Servers and list resource managers</td>
<td>Ip2ListServers()</td>
<td>Not available</td>
<td>DKRMConfigurationMgmtICM.listResourceMgrs()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Connecting to content servers</td>
</tr>
<tr>
<td>Event log</td>
<td>Ip2WriteHistoryEvent()</td>
<td>Not available</td>
<td>DKEventMgmtICM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Planning a Content Manager</td>
</tr>
<tr>
<td>Update password for a session</td>
<td>Ip2ModifyUser()</td>
<td>DKDatastoreDL.changePassword()</td>
<td>dkDatastore.changePassword()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Connecting to content servers</td>
</tr>
<tr>
<td>Register user exits for a session</td>
<td>Ip2SetUserExits()</td>
<td>Not available</td>
<td>Not available</td>
</tr>
<tr>
<td>Start a transaction</td>
<td>Ip2StartTransaction()</td>
<td>DKDatastoreDL.startTransaction()</td>
<td>DKDatastoreICM.startTransaction()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Processing transactions</td>
</tr>
<tr>
<td>Commit</td>
<td>Ip2EndTransaction(OIM_COMMIT)</td>
<td>DKDatastoreDL.commit()</td>
<td>DKDatastoreICM.commit()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Processing transactions</td>
</tr>
<tr>
<td>Roll back</td>
<td>Ip2EndTransaction(OIM_ROLLBACK)</td>
<td>DKDatastoreDL.rollback()</td>
<td>DKDatastoreICM.rollback()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Processing transactions</td>
</tr>
</tbody>
</table>

Table E-2  API migration table for working with items

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check in items</td>
<td>Ip2CheckInItem()</td>
<td>DKDatastoreDL.checkIn()</td>
<td>DKDatastoreICM.checkIn()</td>
</tr>
<tr>
<td>Check out items</td>
<td>Ip2CheckOutItem()</td>
<td>DKDatastoreDL.checkOut()</td>
<td>DKDatastoreICM.checkOut()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td>Processing transactions</td>
</tr>
</tbody>
</table>
### Table E-2  API migration table for working with items

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine if item is checked out</td>
<td>Determine if item is checked out</td>
<td>Determine if item is checked out</td>
<td>Determine if item is checked out</td>
</tr>
<tr>
<td></td>
<td>SimLibGetItemInfo()</td>
<td>DKDatastoreDL.isCheckedOut()</td>
<td>DKDatastoreICM.isCheckedOut()</td>
</tr>
<tr>
<td>Create (import) a new item</td>
<td>SimLibCreateItem()</td>
<td>DKDDO.add()</td>
<td>DKDDO.add()</td>
</tr>
<tr>
<td></td>
<td>SimLibLoadMediaObject()</td>
<td>DKDatastoreDL.addObject(ddo)</td>
<td>DKDatastoreICM.addObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibStoreNewObject()</td>
<td>DKDatastoreExtICM.addObject()</td>
<td>DKDatastoreExtICM.addObject()</td>
</tr>
<tr>
<td></td>
<td>SimLibCreateItemPartExtSrch()</td>
<td>Related information: Working with DDO Importing XML Documents</td>
<td>Related information: Working with DDO Importing XML Documents</td>
</tr>
<tr>
<td></td>
<td>SimLibInvokeSearchEngine()</td>
<td>Related information: Working with DDO Importing XML Documents</td>
<td>Related information: Working with DDO Importing XML Documents</td>
</tr>
<tr>
<td></td>
<td>SimLibAddFolderItem()</td>
<td>Related information: Working with DDO Importing XML Documents</td>
<td>Related information: Working with DDO Importing XML Documents</td>
</tr>
<tr>
<td>Retrieve items</td>
<td>SimLibGetIndexClassView()</td>
<td>DKDDO.retrieve()</td>
<td>DKDDO.retrieve()</td>
</tr>
<tr>
<td></td>
<td>SimLibSetIndexClassView()</td>
<td>DKDatastoreDL.retrieveObject(ddo)</td>
<td>DKDatastoreICM.retrieveObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibGetItemSnapshot()</td>
<td>DKDatastoreExtICM.retrieveObject(ddo)</td>
<td>DKDatastoreExtICM.retrieveObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibGetAffiliatedTOC()</td>
<td>DKDDO.retrieve()</td>
<td>DKDDO.retrieve()</td>
</tr>
<tr>
<td></td>
<td>SimLibGetTOC()</td>
<td>Related information: Working with DDO Importing XML Documents</td>
<td>Related information: Working with DDO Importing XML Documents</td>
</tr>
<tr>
<td></td>
<td>Ip2CloseTOC()</td>
<td>Related information: Working with DDO Importing XML Documents</td>
<td>Related information: Working with DDO Importing XML Documents</td>
</tr>
<tr>
<td>Retrieve a part by name</td>
<td>SimLibOpenByUniqueName()</td>
<td>DKDDO.retrieveFormOverlay()</td>
<td>DKDDO.retrieveFormOverlay()</td>
</tr>
<tr>
<td></td>
<td>SimLibSeekObject()</td>
<td>DKDatastoreICM.createDDO()</td>
<td>DKDatastoreICM.createDDO()</td>
</tr>
<tr>
<td></td>
<td>SimLibReadObject()</td>
<td>DKDDO.retrieve()</td>
<td>DKDDO.retrieve()</td>
</tr>
<tr>
<td>Set or modify attributes in a DDO</td>
<td>Not Available in Folder Manager</td>
<td>DKDDO.setData()</td>
<td>DKDDO.setData()</td>
</tr>
<tr>
<td>Update items</td>
<td>SimLibDeleteItemPartExtSrch()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibDeleteObject()</td>
<td>DKDatastoreDL.updateObject(ddo)</td>
<td>DKDatastoreDL.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibInvokeSearchEngine()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibCreateItemPartExtSrch()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibLoadMediaObject()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibStoreNewObject()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibStoreObject()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibOpenItemAttr()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibWriteAttr()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibCloseAttr()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibAddFolderItem()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibRemoveFolderItem()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
<tr>
<td></td>
<td>SimLibUpdateFolderItem()</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
<td>DKDatastoreExtICM.updateObject(ddo)</td>
</tr>
<tr>
<td></td>
<td>SimLibUpdateObject()</td>
<td>DKDDO.update()</td>
<td>DKDDO.update()</td>
</tr>
</tbody>
</table>

Appendix E. API migration tables for Content Manager 671
<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete items</td>
<td>SimLibDeleteItemPartExt Srch() SimLibDelete Object()</td>
<td>DKDDO.del() DKDatastoreDL. deleteObject(ddo)</td>
<td>DKDDO.del() DKDatastoreICM. deleteObject(ddo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Working with DDOs Working with XDOs</td>
<td>Related information: Working with DDOs Working with XDOs</td>
</tr>
<tr>
<td>Create parts or resource items (such as notes or annotations)</td>
<td>SimLibLoadMediaObject() SimLibStoreNewObject() SimLibStoreObject() SimLibCreateItemPartExt Srch() SimLibInvokeSearch Engine()</td>
<td>DKBlobDL.add()</td>
<td>DKBlobICM.add() Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
<tr>
<td>Retrieve parts or resource items</td>
<td>SimLibGetItemAffiliated TOC() SimLibQueryObject() SimLibOpenObject() Ip2QueryObjectAccess() SimLibCloseObject()</td>
<td>DKBlobDL.retrieve()</td>
<td>DKBlobICM.retrieve() Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
<tr>
<td>Update parts or resource items</td>
<td>SimLibUpdatePartExt Srch() SimLibInvokeSearch Engine()</td>
<td>DKBlobDL.update()</td>
<td>DKBlobICM.update() Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
<tr>
<td>Delete parts or resource items</td>
<td>SimLibDeleteItemPartExt Srch() SimLibDelete Object()</td>
<td>DKBlobDL.del()</td>
<td>DKBlobICM.del() Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
<tr>
<td>Import an object from a file</td>
<td>SimLibCatalogObject() SimLibCreateItemPartExt Srch() SimLibStoreObject() SimLibStoreNewObject() SimLibLoadMediaObject()</td>
<td>DKBlobDL.add(fileName)</td>
<td>DKBlobICM.add(fileName) Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
<tr>
<td>Export a part into a file</td>
<td>SimLibGetItemAffiliated TOC() SimLibOpenObject() Ip2QueryObjectAccess() SimLibCloseObject()</td>
<td>DKBlobDL.retrieve(fileName)</td>
<td>DKBlobICM.retrieve(fileName) Related information: Working with XDOs Creating and using the DKParts attribute Using XDOs</td>
</tr>
</tbody>
</table>
### Table E-2  API migration table for working with items

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add items into a folder</td>
<td>SimLibAddFolderItem()</td>
<td>DKFolder.addMember()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreDL.addFolderItem(folder, member)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreExtICM.addToFolder()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Creating custom content server connectors</td>
</tr>
<tr>
<td>Remove items from a folder</td>
<td>SimLibRemoveFolderItem()</td>
<td>DKFolder.removeMember()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreDL.removeFolderItem(folder, mbr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreExtICM.removeFromFolder()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Creating custom content server connectors</td>
</tr>
<tr>
<td>Add parts to an item, or link items to resource items</td>
<td>SimLibLoadMediaObject() SimLibStoreNewObject() SimLibStoreObject() SimLibCreateInfoPartExtSrch() SimLibInvokeSearchEngine()</td>
<td>DKParts.addMember(doc, part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreExtICM.addLink(link)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Working with XDOs Creating and using the DKParts attribute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating a Content Manager application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using XDOs</td>
</tr>
<tr>
<td>Remove parts from an item, or unlink items from resource items</td>
<td>SimLibDeleteItemPartExtSrch() SimLibDeleteObject()</td>
<td>DKParts.removeMember(doc, part)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreExtICM.removeLink(link)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Update SMS information for a part</td>
<td>SimLibChangeObjectSMS()</td>
<td>DKBlobDL.setExtension()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKBlobDL.changeStorage()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKStorageManagerInfoDL class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Working with XDOs Using XDOs</td>
</tr>
<tr>
<td>Prefetch a part</td>
<td>SimLibOpenObject(GET_IT_PREFETCH)</td>
<td>DKBlobDL.retrieve() with option GET_IT_PREFETCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
<tr>
<td>Define links between items</td>
<td>Not available</td>
<td>Not available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKLinkCollection.addElement()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKLinkCollection.addMember()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
</tbody>
</table>
### Table E-2  API migration table for working with items

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete links between items</td>
<td>Not available</td>
<td>Not available</td>
<td>DKLinkCollection.removeElement() DKLinkCollection.removeMember() DKDatastoreExtICM. removeLink(link)</td>
</tr>
<tr>
<td>Retrieve links</td>
<td>Not available</td>
<td>Not available</td>
<td>DKDDO.retrieve() with option DKConstant.DK_CM_CONTENT_LINKS_OUTBOUND + DKConstant.DK_CM_CONTENT_LINKS_INBOUND</td>
</tr>
<tr>
<td>Move objects</td>
<td>SimLibOpenItemAttr()</td>
<td>DKDatastoreDL.moveObject()</td>
<td>DKDatastoreICM.moveObject() DKDDO.retrieve() DKConstant.DK_CM_CONTENT_LINKS_OUTBOUND DKConstant.DK_CM_CONTENT_LINKS_INBOUND Related information: Creating a Content Manager application</td>
</tr>
</tbody>
</table>

### Table E-3  API migration table for data modeling

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create index class or item type</td>
<td>Ip2CreateClass()</td>
<td>Not available</td>
<td>DKItemTypeDefICM.add() or DKComponentTypeDefICM.del() DKDatastoreICM.listEntities() Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>List index classes or item types</td>
<td>SimLibListClasses()</td>
<td>DKDatastoreDL.listEntities()</td>
<td>DKDatastoreICM.listEntities() DKDatastoreICM.listEntities() Related information: Connecting to content servers</td>
</tr>
<tr>
<td>Update index classes or item types</td>
<td>Not available in FM.</td>
<td>Not available</td>
<td>DKItemTypeDefICM.update() or DKComponentTypeDefICM.update() DKDatastoreICM.listEntities() Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Task</td>
<td>Folder Manager</td>
<td>DL Connector</td>
<td>ICM Connector</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Delete index classes or item types</td>
<td>Ip2DeleteIndex()</td>
<td>Not available</td>
<td>DKItemTypeDefICM.del() or DKComponentTypeDefICM.del()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Define attributes</td>
<td>Ip2CreateAttr()</td>
<td>Not available</td>
<td>DKAttrDefICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>List index class or item type attributes</td>
<td>SimLibGetAttrInfo()</td>
<td>DKDatastoreDL.listAttrs()</td>
<td>DKDatastoreICM.listAttrs()</td>
</tr>
<tr>
<td></td>
<td>Ip2ListAttrs()</td>
<td>DKDatastoreDL.listEntityAttrs(entityName)</td>
<td>DKDatastoreICM.listEntityAttrs(entityName)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Connecting to content servers</td>
</tr>
<tr>
<td>Update attribute definitions</td>
<td>Ip2ModifyAttr()</td>
<td>Not available</td>
<td>DKAttrDefICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Delete attribute definitions</td>
<td>Ip2DeleteAttr()</td>
<td>Not available</td>
<td>DKAttrDefICM.delete()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Add an attribute to an item type</td>
<td>Not available</td>
<td>Not available</td>
<td>DKItemTypeDef.add(attribute)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Creating a Content Manager application</td>
</tr>
<tr>
<td>Add index class or item type views</td>
<td>Ip2CreateClass()</td>
<td>Not available</td>
<td>DKItemTypeViewDefICM.add() or DKComponentTypeViewDefICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with items</td>
</tr>
<tr>
<td>Update index class or item type views</td>
<td>Not available</td>
<td>Not available</td>
<td>DKItemTypeViewDefICM.update() or DKComponentTypeViewDefICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Deleting item type views</td>
</tr>
<tr>
<td>Delete index-class or item type views</td>
<td>Ip2DeleteIndex()</td>
<td>Not available</td>
<td>DKItemTypeViewDefICM.del() or DKComponentTypeViewDefICM.del()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Deleting item type views</td>
</tr>
</tbody>
</table>
Table E-4  API migration table for user and authorization management

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add user and user group definitions</td>
<td>Ip2AddUser() Ip2AddGroup()</td>
<td>DKUserMgmtDL.add()</td>
<td>DKUserMgmtICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining users</td>
</tr>
<tr>
<td>Update user and user group definitions</td>
<td>Ip2ModifyUser() Ip2ModifyGroup()</td>
<td>DKUserMgmtDL.update()</td>
<td>DKUserMgmtICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining users</td>
</tr>
<tr>
<td>Delete user and user group definitions</td>
<td>Ip2DeleteUser() Ip2DeleteGroup()</td>
<td>DKUserMgmtDL.del()</td>
<td>DKUserMgmtICM.del()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining users</td>
</tr>
<tr>
<td>Add association of user to user group</td>
<td>Ip2AddUserToGroup()</td>
<td>Not available</td>
<td>DKUserGroupDefICM.addUser()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining users</td>
</tr>
<tr>
<td>Remove association of user to user group</td>
<td>Ip2RemoveUserFromGroup()</td>
<td>Not available</td>
<td>DKUserGroupDefICM.removeUser()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining users</td>
</tr>
<tr>
<td>Define privilege, set, or group</td>
<td>Ip2CreatePrivSet()</td>
<td>Not available</td>
<td>DKAuthorizationMgmtICM.createPrivilege()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DKAuthorizationMgmtICM.createPrivilegeGroup()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DKAuthorizationMgmtICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining privileges</td>
</tr>
<tr>
<td>Update privilege, set, or group</td>
<td>Ip2ModifyPrivSet()</td>
<td>Not available</td>
<td>DKAuthorizationMgmtICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining privileges</td>
</tr>
<tr>
<td>Delete privilege, set, or group</td>
<td>Ip2DeletePrivSet()</td>
<td>Not available</td>
<td>DKAuthorizationMgmtICM.del()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining privileges</td>
</tr>
<tr>
<td>List privileges of a user</td>
<td>Ip2ListUser()</td>
<td>DKAccessControlDL.listPrivilege()</td>
<td>TBD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Defining privileges</td>
</tr>
<tr>
<td>Add ACL</td>
<td>Ip2UpdateAccessList()</td>
<td>DKAccessControlDL.addAccessControlList()</td>
<td>DKAuthorizationMgmtICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Working with access control lists</td>
</tr>
<tr>
<td>Update ACL</td>
<td>Ip2UpdateAccessList()</td>
<td>DKAccessControlDL.updateAccessControlList()</td>
<td>DKAuthorizationMgmtICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Working with access control lists</td>
</tr>
</tbody>
</table>
### Table E-4  API migration table for user and authorization management

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete ACL</td>
<td>Ip2UpdateAccessList()</td>
<td>DKAccessControlDL.deleteAccessControlList()</td>
<td>DKAuthorizationMgmtICM.del()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with access control lists</td>
</tr>
</tbody>
</table>

### Table E-5  API migration table for MIME and configuration management

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define MIME type definition</td>
<td>Ip2AddContentClass2()</td>
<td>DKDatastoreAdminDL.addContentDef()</td>
<td>DKMimeTypeMgmtICM.add()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
<tr>
<td>Update MIME type definition</td>
<td>Ip2ModifyContentClass2()</td>
<td>DKDatastoreAdminDL.updateContentDef()</td>
<td>DKMimeTypeMgmtICM.update()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
<tr>
<td>Delete MIME type definition</td>
<td>Ip2DeleteContentClass()</td>
<td>DKDatastoreAdminDL.deleteContentDef()</td>
<td>DKMimeTypeMgmtICM.delete()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
<tr>
<td>Add SMS collection</td>
<td>Ip2SMSCreateEntity(SMS_COLLECTION)</td>
<td>Not available</td>
<td>DKRMConfigurationMgmtICM.addSMSCollection()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
<tr>
<td>Delete SMS collection</td>
<td>Ip2SMSDeleteEntity(SMS_COLLECTION)</td>
<td>Not available</td>
<td>DKRMConfigurationMgmtICM.delSMSCollection()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Working with objects</td>
</tr>
</tbody>
</table>

### Table E-6  API migration table for workflow

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start a process</td>
<td>Ip2StartWorkFlow()</td>
<td>DKWorkFlowServiceDL.startWorkFlowItem(itemId, itemIDWF, itemIDWB, overload, priority)</td>
<td>DKDocRoutingServiceICM.startProcess(process_name, itemId, priority)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td>Terminate a process</td>
<td>Ip2CompleteWorkFlow()</td>
<td>DKWorkFlowServiceDL.completeWorkFlowItem(itemId)</td>
<td>DKDocRoutingServiceICM.terminateProcess(workpacket_pid)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
</tbody>
</table>
Table E-6  API migration table for workflow

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue a process</td>
<td>Ip2RouteWipItem()</td>
<td>DKWorkFlowServiceDL. routeWipItem(itemID, itemIDWB, overload, priority)</td>
<td>DKDocRoutingServiceICM. continueProcess(workpacket_pid, selection)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td>Suspend a process</td>
<td>Ip2SuspendItem()</td>
<td>Not available</td>
<td>DKDocRoutingServiceICM. suspendProcess(workpackage_pid, suspend_unit, duration, resume_list)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td>Resume a process</td>
<td>Ip2ActivateItem()</td>
<td>Not available</td>
<td>DKDocRoutingServiceICM. resumeProcess(workpacket_pid)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td>Retrieve the next work item (work list)</td>
<td>Ip2GetNextWorkBasketItem()</td>
<td>DKWorkBasketDL. getNextHighPriorityItem()</td>
<td>DKDocRoutingServiceICM. getNextWorkPackagePidString(worklist)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DKDocRoutingServiceICM. getNextWorkPackage(worklist)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td>List all of the work from a work list</td>
<td>SimLibGetTOC()</td>
<td>DKWorkBasketDL. listItemIDs(wip_status)</td>
<td>DKDocRoutingServiceICM. listWorkPackagePidStrings(worklist)</td>
</tr>
<tr>
<td></td>
<td>Ip2CloseTOC()</td>
<td>DKWorkBasketDL. listWorkManagementInfos(wip_status)</td>
<td>Related information: Routing a document through a process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKWorkFlowServiceDL. listWorkFlowIDs()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKWorkFlowServiceDL. listWorkFlows()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKWorkFlowServiceDL. listWorkFlows()</td>
<td>Related information: Routing a document through a process</td>
</tr>
</tbody>
</table>
## Table E-6  API migration table for workflow

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>List work nodes</td>
<td>Ip2ListWork Baskets()</td>
<td>DKWorkFlowServiceDL.</td>
<td>DKDocRoutingServiceMgmtICM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>listWorkBasketIDs()</td>
<td>listWorkNodeNames()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKWorkFlowServiceDL.</td>
<td>DKDocRoutingServiceMgmtICM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>listWorkBaskets()</td>
<td>listWorkNodes()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDocRoutingServiceMgmtICM.</td>
<td>Related information:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>listWorkNodeNames()</td>
<td>Routing a document through a process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDocRoutingServiceMgmtICM.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>listWorkLists()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDocRoutingServiceMgmtICM.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>listWorkLists()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Routing a document through a process</td>
<td></td>
</tr>
</tbody>
</table>

## Table E-7  API migration table for Parametric Search

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search for items that match criteria</td>
<td>SimLibSearch()</td>
<td>DKDatastoreDL.evaluate()</td>
<td>DKDatastoreICM.evaluate()</td>
</tr>
<tr>
<td></td>
<td>SimLibGetItemInfo()</td>
<td>DKDatastoreDL.execute()</td>
<td>DKDatastoreICM.execute()</td>
</tr>
<tr>
<td></td>
<td>SimLibSetIndex</td>
<td>DKDatastoreDL.</td>
<td>DKDatastoreICM.</td>
</tr>
<tr>
<td></td>
<td>ClassView()</td>
<td>executeWithCallback()</td>
<td>executeWithCallback()</td>
</tr>
<tr>
<td></td>
<td>SimLibGetItemSnapshot()</td>
<td>dkResultSetCursor.fetchNext()</td>
<td>dkResultSetCursor.fetchNext()</td>
</tr>
<tr>
<td></td>
<td>SimLibGetAffiliatedTOC()</td>
<td>dkResultSetCursor.fetchObject()</td>
<td>dkResultSetCursor.fetchObject()</td>
</tr>
<tr>
<td></td>
<td>Ip2CloseTOC()</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Related information:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Querying a content server</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using the result set cursor</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Querying collections</td>
<td></td>
</tr>
</tbody>
</table>

## Table E-8  API migration table for Text Search

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect to a text search server disconnect</td>
<td>Not available</td>
<td>DKDatastoreTS.connect()</td>
<td>Connection is implicitly and automatically performed by the system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreTS.disconnect()</td>
<td></td>
</tr>
<tr>
<td>Perform a text search</td>
<td>Not available</td>
<td>DKDatastoreTS.evaluate():</td>
<td>Integrated with parametric query</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreTS.execute()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DKDatastoreTS.executeWithCallback()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dkResultSetCursor.fetchNext()</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dkResultSetCursor.fetchObject()</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Folder Manager</td>
<td>DL Connector</td>
<td>ICM Connector</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------</td>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Add and index a part or a resource item</td>
<td>SimLibLoadMediaObject() SimLibStoreNewObject() SimLibStoreObject() SimLibCreateItemPartExtSrch() SimLibInvokeSearchEngine()</td>
<td>DKBlobDL.add()</td>
<td>DKTextICM.add()</td>
</tr>
<tr>
<td>Index an existing (text) part</td>
<td>SimLibIndexPartExtSrch() SimLibInvokeSearchEngine()</td>
<td>DKBlobDL.setToBeIndexed()</td>
<td>Use DKDDO.setData() on TIEFlag attribute: 1 to turn text search on 0 to turn text search off Then, perform a DKDDO.update()</td>
</tr>
<tr>
<td>Create a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.createIndex()</td>
<td>Use setTextSearchable (true) in DKAttrDefICM and DKItemTypeDefICM classes. Text index properties are assigned by default, or can be specified through the DKTextIndexDefICM class.</td>
</tr>
<tr>
<td>Delete a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.deleteIndex()</td>
<td>This function is implicitly and automatically performed by the system when an item type is deleted. Or, use setTextSearchable (false) on the attribute and then perform an update.</td>
</tr>
<tr>
<td>Clear all indexed terms from a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.clearIndex()</td>
<td>Use setTextSearchable(false) on the attribute and then perform an update.</td>
</tr>
<tr>
<td>Get information on a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.getIndexInformation()</td>
<td>Provided by DKTextIndexDefICM class</td>
</tr>
<tr>
<td>Get indexing function status of a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.getIndexFunctionStatus()</td>
<td>Not exposed</td>
</tr>
<tr>
<td>Set indexing function for a text search index</td>
<td>Not available</td>
<td>DKDatastoreTS.setIndexFunctionStatus()</td>
<td>Not exposed</td>
</tr>
</tbody>
</table>
### Table E-8  API migration table for Text Search

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start the text indexing process</td>
<td>Not available</td>
<td>DKDatastoreTS. startUpdateIndex()</td>
<td></td>
<td>This process is performed either by the system based on update settings in the DKTextIndexDefICM class or programmatically through DKDatastoreDefICM.updateTextIndexes (yourComponentTypeID) and DKDatastoreDefICM.reorgTextIndexes (yourComponentTypeID).</td>
</tr>
</tbody>
</table>

### Table E-9  API migration table for combined search

<table>
<thead>
<tr>
<th>Task</th>
<th>Folder Manager</th>
<th>DL Connector</th>
<th>ICM Connector</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search for items that match parametric and text criteria</td>
<td>SimLibSetIndex ClassView() SimLibSearch() SimGetItemInfo() SimLibGet AffiliatedTOC() SimLibGetTOC()</td>
<td>DKDatastoreDL.evaluate() DKDatastoreDL.execute() DKDatastoreDL.executeWithCallback() dkResultSetCursor.fetchNext() dkResultSetCursor.fetchObject() (The query is a combination of parametric and text query.)</td>
<td>DKDatastoreICM.evaluate() DKDatastoreICM.execute() DKDatastoreICM.executeWithCallback() dkResultSetCursor.fetchNext() dkResultSetCursor.fetchObject() Content Manager Version 8.1 supports integrated parametric and text query, in other words, you can mix parametric and text criteria in a single query. Related information: Querying a content server Using the result set cursor Querying collections</td>
<td>Not available</td>
</tr>
<tr>
<td>Search for items that match parametric, text, and image criteria</td>
<td>SimLibSetIndex ClassView() SimLibSearch() SimGetItemInfo() SimLibGet AffiliatedTOC() SimLibGetTOC()</td>
<td>DKDatastoreDL.evaluate() DKDatastoreDL.execute() DKDatastoreDL.executeWithCallback() dkResultSetCursor.fetchNext() dkResultSetCursor.fetchObject() (The query is a combination of parametric, text, and image query.)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuration and log files

This appendix lists the configuration files and log files used in a Content Manager system.

**Configuration files**

In Content Manager V8.3, an improvement is that most of the configuration files have been consolidated into one location. The common configuration file location is the cmgmt directory in the working directory. The Content Manager working directory is located in the following directory:

- **Windows**: %IBMCMROOT% (default)
- **UNIX**: Home directory of the new system user ibmcmadm. For example: /home/ibmcmadm/
Figure F-1 lists the structure of the common configuration file directory.

![Common Configuration Structure](image)

Each subdirectory corresponds to a component of Content Manager as described below:

- /connectors: The subdirectory for configuration files that are related to connectors.
- /sa: The subdirectory for configuration files that are related to the system administration client.
- /beans: The subdirectory for configuration files that are related to beans and Web services.
- /eClient: The subdirectory for configuration files that are related to eClient.
- /ls: The subdirectory for configuration files that are related to Library Server. The structure is /ls/<LSDBName>;
  Where <LSDBName> is the name of the Library Server database.
- /rm: The subdirectory for configuration files that are related to Resource Manager. The structure is /rm/<WASNode>/<WASApp> Where <WASNode> is the node name of WebSphere Application Server and <WASApp> is the application name of the Resource Manager;
- /icmclient: The subdirectory for configuration files that are related to Client for Windows. The structure is /icmclient/<SysUsr> Where <SysUsr> is the name of the system user
Table F-1 lists the configuration files that are used by Content Manager.

<table>
<thead>
<tr>
<th>File name</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
</table>
|.cmbcmenv.properties         | 1. `<IBMCMROOT>/cmgmt`  
                           | 2. `<WAS_ROOT>/installed Apps/<WASNode>/cmrrm.ear/cmrrm.war/WEB-INF/classes/com/ibm/mm/icmrmm` | 1. Defines the location that `<CMCFGDIR>` points to  
                               | 2. LDAP information                                                          |
| ibmcmconfig.properties      | `<IBMCMROOT>/cmgmt/`                                                  | Defines the location that `<IBMCMWorkingDirectory>` points to                               |
| ICMClientLog.ini            | `<IBMCMROOT>/cmgmt/icmclient/<SysUsr>/`                                | Defines tracing for client components                                                        |
| cmbicmenv.ini               | `<WorkingDirectory>/cmgmt/connectors/`  
                               | (or where specified in `<CMCFGDIR>`)                                                      | Contains the encrypted DB2 connection information for the Library Server(s) defined in cmbicmsrvs.ini |
| cmbicmsrvs.ini              | `<WorkingDirectory>/cmgmt/connectors/`  
                               | (or where specified in `<CMCFGDIR>`)                                                      | Lists available Library Server(s) and their connection information.  
                               | For detail, please read DB2 Content Manager v8.3 manual  
                               | “System Administration Guide”, “Database connection parameter files” in chapter 3 |
| cmbcs.ini                   | `<WorkingDirectory>/cmgmt/connectors/`  
                               | (or where specified in `<CMCFGDIR>`)                                                      | Contains information about whether to use the local or remote (RMI) connectors |
| cmbclient.ini               | `<WorkingDirectory>/cmgmt/connectors/`  
                               | (or where specified in `<CMCFGDIR>`)                                                      | Contains information about the remote RMI server parameters |
| cmblogconfig.properties     | `<WorkingDirectory>/cmgmt/connectors/`  
<pre><code>                           | (or where specified in `&lt;CMCFGDIR&gt;`)                                                      | Used to control the type of log manager used and the configuration specific to each type of log manager |
</code></pre>
<p>| icmrm_logging.xml           | <code>&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA pp&gt;</code>                       | Contains configuration information about the general Resource Manager log |</p>
<table>
<thead>
<tr>
<th>File name</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icmrm_migrator</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the migrator log</td>
</tr>
<tr>
<td>_logging.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>icmrm_purger</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the purger log</td>
</tr>
<tr>
<td>_logging.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>icmrm_stager</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the stager log</td>
</tr>
<tr>
<td>_logging.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>icmrm_replicator</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the replicator log</td>
</tr>
<tr>
<td>_logging.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>icmrm_validator_log</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the validation utilities log</td>
</tr>
<tr>
<td>g.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>icmrm.Asyncr</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/rm/&lt;WASNode&gt;/&lt;WASA</td>
<td>Contains configuration information about the asynchronous recovery utility log</td>
</tr>
<tr>
<td>_logging.xml</td>
<td>pp&gt;/</td>
<td></td>
</tr>
<tr>
<td>cmadmin.properties</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/sa/</td>
<td>Indicates the number and type of servers being administered by the system administration client</td>
</tr>
<tr>
<td>icm.properties</td>
<td>`&lt;IBMCMROOT&gt;/lib/</td>
<td>Contains Content Manager administration specifics such as tracing levels</td>
</tr>
<tr>
<td>cmbpool.ini</td>
<td>`&lt;WorkingDirectory&gt;/cmgmt/connector</td>
<td>Define properties for JDBC database connection pool</td>
</tr>
</tbody>
</table>

Where:

- `<IBMCMROOT>` is the directory where DB2 Content Manager is installed. For example, E:\IBM\db2cmv8.
- `<CMCFGDIR>` is the location for Content Manager configuration files, as defined in cmbcmenv.properties.
- `<WASNode>` is the node name of WebSphere Application Server.
- `<WASApp>` is the application name of the Resource Manager.
- `<SysUsr>` is the name of the system user.
- `<WorkingDirectory>` is the Content Manager working directory located in the following directory:
  - Windows: `%IBMCMROOT%` (default)
  - UNIX: Home directory of the new system user ibmcmadm. For example: `/home/ibmcmadm/

**Log files**

In Content Manager V8.3, most of the log files have been consolidated into one location. The common log location is the log directory in the working directory. The structure of the log directory is similarly to the configuration file directory.

Figure F-2 list the structure of common log directory.

![Common Log Structure](image)

*Figure F-2  Common Log Structure*

Each subdirectory corresponds to a component of Content Manager as described below:
- `/connectors`: The subdirectory for log files that are related to connectors.
- `/sa`: The subdirectory for log files that are related to the system administration client.
- `/beans`: The subdirectory for log files that are related to beans and Web services.
- `/eClient`: The subdirectory for log files that are related to eClient.
- /ls: The subdirectory for log files that are related to Library Server. The structure is /ls/<LSDBName>

Where <LSDBName> is the name of the Library Server database

- /rm: The subdirectory for log files that are related to Resource Manager. The structure is /rm/<WASNode>/<WASApp>

Where <WASNode> is the node name of WebSphere Application Server and <WASApp> is the application name of the Resource Manager

- /icmclient: The subdirectory for log files that are related to Client for Windows. The structure is /icmclient/<SysUsr>

Where <SysUsr> is the name of the system user

Table F-2 lists the log files that are used by Content Manager:

Table F-2  Log file summary

<table>
<thead>
<tr>
<th>File name</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmininstall.log</td>
<td>&lt;IBMCMROOT&gt;/log</td>
<td>Installation log</td>
</tr>
<tr>
<td>ICMSERVER.LOG</td>
<td>&lt;WorkingDirectory&gt;/log /ls/&lt;LSDBName&gt; (or where specified in the system administration client)</td>
<td>General Library Server log information</td>
</tr>
<tr>
<td>&lt;SysUsr&gt;.cmadmin.log</td>
<td>&lt;WorkingDirectory&gt;/log /sa/</td>
<td>Common system administration client log information</td>
</tr>
<tr>
<td>&lt;SysUsr&gt;.dklog.log</td>
<td>&lt;WorkingDirectory&gt;/log /connectors/</td>
<td>Application log file</td>
</tr>
<tr>
<td>ICMClient.err</td>
<td>&lt;WorkingDirectory&gt;/log /icmclient/&lt;SysUsr&gt;/</td>
<td>Log for Client for Windows</td>
</tr>
<tr>
<td>&lt;SysUsr&gt;.beans.log</td>
<td>&lt;WorkingDirectory&gt;/log /beans/</td>
<td>Beans log file</td>
</tr>
<tr>
<td>icmrm.logfile</td>
<td>&lt;WorkingDirectory&gt;/log /rm/&lt;WASNode&gt;/&lt;WASApp&gt; (or where specified in the xml configuration file)</td>
<td>Contains information about the general Resource Manager log</td>
</tr>
<tr>
<td>File name</td>
<td>Location</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>icmrm.migrator.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the migrator log</td>
</tr>
<tr>
<td>icmrm.purger.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the purger log</td>
</tr>
<tr>
<td>icmrm.stager.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the stager log</td>
</tr>
<tr>
<td>icmrm.replicator.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the replicator log</td>
</tr>
<tr>
<td>icmrm.validator.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the validation utilities log</td>
</tr>
<tr>
<td>icmrm.asyncr.logfile</td>
<td><code>&lt;WorkingDirectory&gt;/log/rm/&lt;WASNode&gt;/&lt;WASApp&gt;/</code> (or where specified in the xml configuration file)</td>
<td>Contains information about the asynchronous recovery utility log</td>
</tr>
</tbody>
</table>
Where:

- `<IBMCMROOT>` is the directory where DB2 Content Manager is installed. For example, E:\IBM\db2cmv8
- `<CMCFGDIR>` is the location for DB2 Content Manager configuration files, as defined in cmbcmenv.properties.
- `<WASNode>` is the node name of Websphere Application Server
- `<WASApp>` is the application name of the Resource Manager
- `<SysUsr>` is the name of the system user
- `<LSDBName>` is the name of the Library Server database
- `<WorkingDirectory>` is the Content Manager working directory located in the following directory:
  - Windows: `%IBMCMROOT%` (default)
  - UNIX: Home directory of the new system user ibmcadm. For example: /home/ibmcadm/
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 693. Note that some of the documents referenced here may be available in softcopy only.

- *DB2 Content Manager for z/OS: Implementation, Installation, and Migration*, SG24-6476
- *Performance Tuning for Content Manager*, SG24-6949
- *Content Manager Backup/Recovery and High Availability: Strategies, Options, and Procedures*, SG24-7063
- *Implementing Web Application Using Information Integrator for Content and OnDemand Web Enablement Kit*, SG24-6338
- *eClient 101 Customization and Integration*, SG24-7063
- *Content Manager Version 8.1 Migration Guide for Multiplatforms*, SG24-6877
- *IBM Tivoli Storage Manager Version 5.1 Technical Guide*, SG24-6554
- *IBM Tivoli Storage Management Concepts*, SG24-4877

Other publications

These publications are also relevant as further information sources:

- *IBM Content Manager for Multiplatforms - Revised Installation Steps for Windows*
- *IBM Content Manager for Multiplatforms - Revised Installation Steps for AIX*
- *IBM Content Manager for Multiplatforms - Revised Installation Steps for Sun Solaris*
- *IBM DB2 Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332
- "IBM DB2 Content Manager for Multiplatforms System Administration Guide, SZ27-1335"
- "IBM DB2 Content Manager Enterprise Edition / IBM DB2 Content Manager for z/OS Client V8.3: Client for Windows Programming Reference, SC27-1337"
- "IBM Content Manager for Multiplatforms - Migrating to Content Manager Version 8, SC27-1343"
- "IBM Content Manager for Multiplatforms - Upgrade from Version 8.1 to Version 8.2 Guide"
- "IBM Content Manager for Multiplatforms - Messages and Codes, SC27-1349"
- "IBM Content Manager - Migrating to Content Manager Version 8 for z/OS, GC18-7699"
- "IBM Content Manager - Planning and Installing Your Content Management System for z/OS, GC18-7698"
- "Program Directory for IBM DB2 Content Manager Library Server for z/OS, GI10-8530"
- "Program Directory for IBM DB2 Content Manager Resource Manager for z/OS, GI10-8531"
- "IBM DB2 Universal Database - Message Reference Volume 1, GC09-2978"
- "IBM DB2 Universal Database - Message Reference Volume 2, GC09-2979"
- "IBM DB2 Net Search Extender Administration and User's Guide, SH12-6740"
- "IBM Tivoli Storage Manager for Windows - Quick Start, GC32-0784"
- "IBM Tivoli Storage Manager for AIX - Quick Start, GC32-0770"
- "IBM Tivoli Storage Manager for Windows - Administrator's Guide, GC32-0782"
- "IBM Tivoli Storage Manager for AIX - Administrator's Guide, GC32-0768"
- "IBM Tivoli Storage Manager Administrator's Reference, GC32-0769"
- "IBM Tivoli Storage Manager - Backup-Archive Clients Installation and User's Guide, GC32-0789"
- "IBM DB2 Universal Database - Command Reference, SC09-4828"
- "IBM DB2 Universal Database - System Administration Guide, SC09-4820"
- "DFSMS/MVS V1R5 OAM PISA for Tape Libraries, SC26-3051"
Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM DB2 Content Manager V8.3 Information Center:
  http://publib.boulder.ibm.com/infocenter/cmgmt/v8r3m0
- DB2 Content Management Support page:
  http://www.ibm.com/software/data/cm/cmgr/mp/support.html
- TSM support Web site
- TSM home page documentation:
  http://www.ibm.com/software/tivoli
- IBM Publications Center
  http://www.ibm.com/shop/publications/order
- World Wide Web Consortium (W3C) Web site:
  http://www.w3.org

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

  ibm.com/redbooks

Help from IBM

IBM Support and downloads
  ibm.com/support

IBM Global Services
  ibm.com/services
Symbols
.profiles 616, 627

Numerics
3-tier configuration 288

A
access control 21, 23
access control list 103, 206
access control lists 77, 188, 195
access control user exit routines 212
access data 161
access objects 218
accessing domains 217
accounting log file 423
ACL 83, 103, 206–207
assigning privilege set to 211
Library Server database tables 207
ACL exit routines
compiling 213
ACL Rules for Public 207
ACL user exits
enabling 212
ACLs 196, 293
ACME Marketing 311
architecture 319
data model 324
add() 153
administering domains 216
administration
TSM 240
administration user 230
administrative command routing 241
administrative domain 214
administrative domains 25, 201
ADSM 417–419, 421, 424
advanced queries 180
advanced search 6
advanced text search 170, 175
analyze business operations 286
annotation 58
API 20
API layers available in Content Manager V8 457
API logging 157
APIs 22, 115, 132, 143, 149, 182, 455
available in Content Manager V7 456
need to re-evaluate when porting application 464
application development toolkits 26
Application Programming Interfaces 20, 132
architecture 20, 311
TSM 238
archive 237
archive computer printout 6
ASCII conversion 443
assigning collection to domain 218
assigning privilege set to ACL 211
assigning privilege set to domain 217
assigning Resource Manager to domain 218
assigning user group to domain 217
assigning user to domain 217
assigning users rights 199
associated base part 50
Asynchronous Recovery utilities 485
Asynchronous Recovery Utility 151
Asynchronous Recovery utility 490
attribute 30–31, 72
attribute group 30, 33–34, 40, 73
attribute groups 73
attribute indexes
creating 551
attribute text-searchable 122
attribute type 32
attributes 21, 32–33, 55, 60, 69, 73, 291
authentication 23, 188–189
authorization 23, 195
auto-linking 59
automation 142
AVI video 39
AVT00004 651
AVT00005 651

B
backup 20, 359, 362, 503, 506
backups 237
base 58
basic text search 169, 174
best practices
- performance planning 295
bi-directional association 60
bills 6
binary data 33
Binary Larger Objects 40
bind 437
BLOB 31, 33, 40, 122
BLOBs 150
bottlenecks 545
browser-based client 25
buffer 33
business analysis 285
business operations 286
Business Volume Metrics 290, 544
BVM 544

C
C++ code samples 143
calculating user access rights 196
capacity planning 294
case study scenario 311
categorizing 6
Center 115
central event logging functions 241
character 32
characteristic 31
check-in 152
checkln() 153
checklist 299
check-out 152
checkOut() 153
check-printing 13
child component 30, 36–37, 65, 71–72
child components 35, 38, 43
CICS 595, 647
classification 43
Client for Windows 25
Client for Windows Automation Interface
  porting 461
client option sets 239
client options file 239
CLOB 31, 33, 122
CLOBs 150
cm7bind.err 397
cm8bind.err 397
cm8bind.err. 402
cmbcm81.jar 135
cmbcm81x.dll 135
cmbcm81x.lib 135
cmbcm81xd.dll 135
cmbcm81xd.lib 135
CMBICENV81.bat 645
cmbicm81.jar 135
cmbicm81x.dll 135
cmbicm81x.lib 135
cmbicm81xd.dll 135
cmbicm81xd.lib 135
cmblogconfig.properties 159
cmbsdk81.jar 135
code page 443
code samples 143
collaboration management 3
collection 6, 297
collection mismatch 496
collection point 88
collection points 77
com.ibm.mm.sdk.common 135
com.ibm.mm.sdk.server 135
combined search 22
commit 151
commit() 152–153
Common Secure Interoperability Version 2 226, 228
CommonStore 7
  for Exchange Server 8
  for Lotus Domino 8
  for SAP 8
Commonstore 4
Communication Layer 455
comparison operators 167
component 35
components
  backup 359
ComponentTypeID 58, 60
computer print output 6
configuration 24
  3-tier 288
Content Manager 289
configure SSL for IBM HTTP Server 605
configuring SSL for IBM HTTP Server 617
connect user ID 202
connection
  establish to Library Server database 371
connections
defining for AIX 622
defining for Sun Solaris 634
defining for Windows 610
connectors 24
consideration
  migration 432
considerations
  threading in beans 466
container 58
Contains 61
content 23
Content Manager
  migration 427
content management 3
content management portfolio 4
Content Manager
  database installation customization 550
  installation process for z/OS 635
  installing for z/OS 651
  installing on AIX 620
  installing on Sun Solaris 631
  installing on Windows 608
  migrating from systems earlier than V7 360
  migrating within Windows platform 370
  pinpoint problem 564
Content Manager connectors 133
Content Manager database definitions 651
Content Manager sample code 139
continue a process 80
controlled entities 195
controlled entity 207
copy group 245
CPU 545
CRM 22
cross-platform migration 427
CSlv2 226, 228
Customer Service for Call Center 7
cycle times 483

D
DASD 506
data archive 6
data management API 242
data migration
  overview 360
  preparation 361
  steps 361
data model 20–21, 72, 162, 311
  planning and designing 291
data modeling 29, 132
  case study 322
  key concepts 291
data modeling APIs 145
data modeling entity 30
data protection 237, 248
data stored 161
database 22
  recreate Resource Manager database on AIX 571
  recreate Resource Manager database on Solaris 572
database access passwords
  changing 205
data base authentication 194
data base dump 423
data base environment
  establish 620, 631
data base indexing 68
data base transactions 238
databases
  optimizing 472
dataset 232
date 33
DB2 5, 21
DB2 administration privileges 201
DB2 Client Configuration Assistant 437
DB2 Connect 437
DB2 instance 617, 627
DB2 Performance Expert 553
DB2 Text Information Extender 22
DB2 Universal Database Net Search Extender 164
DB2LOOK 443
DB2MOVE 443
DB2MOVE EXPORT 435
DB2MOVE IMPORT 435
DB2MOVE IMPORT REPLACE_CREATE 436
db2profile 617
DB2RBIND 475
DBADM 198
dbheap 552
DBRM 638
DCB parameters 638
DDL 436, 550
DDO 148
ddos 150
default domains 216
default item type classifications 44
default link type definitions 61
DefaultDomain 216
delete a process 103
deleteObject() 153
deletion reconciliation utility 491
deletion rule of references 66
destage 365
device class
  TSM policy object 246
devices 23
diagnostic 559
differences between Java and C++ APIs 143
digital rights management 3
disaster preparation 237
Disaster Recovery Manager 421
disk space 358
  calculating needed 359
disks 545
distributed searches 5
DK_CM_PARM_END 183
DK_CM_PARM_MAX_RESULTS 183
DK_CM_PARM_PREFETCH_SIZE 183
DK_CM_PARM_RETRIEVE 183
DK_CM_PREPARE_QUERY 183
DKAttrDefICM 130
DKDatastoreDefICM.reorgTextIndexes 129
DKDatastoreICM 144, 150
DKDatastoreICM.add() 153
DKDatastoreICM.checkIn() 153
DKDatastoreICM.checkOut() 153
DKDatastoreICM.commit() 152–153
DKDatastoreICM.deleteObject() 153
DKDatastoreICM.moveObject() 154
DKDatastoreICM.retrieveObject() 153
DKDatastoreICM.rollback() 153
DKDatastoreICM.startTransaction() 152–153
DKDatastoreICM.updateObject() 153
DKFolder 61
DKImageICM 40
DKItemTypeDefICM 130
DKLobICM 41, 152
DKNVPair 183
DKPidICM 465
dkResource 465
DKStreamICM 41
DKTextICM 41, 119
DKTextIndexDefICM 130
DKVideoStreamICM 41
DL Connector
  porting 459
DL connector APIs mapped to ICM connector
DKAccessControlDL
  addAccessControlList 676
deleteAccessControlList 676
listPrivilege 676
updateAccessControlList 676
DKBlobDL
  add 672, 680
  add(fileName) 672
  changeStorage 673
del 672
  retrieve 672–673
  retrieve(fileName) 672
  setExtension 673
  setToBeIndexed 680
  update 672
DKDatastoreDL
  retrieveObject(ddo) 671
DKDatastoreAdminDL
  addContentDef 677
  deleteContentDef 677
  updateContentDef 677
DKDatastoreDL
  addFolderItem(folder, member) 672
  addObject(ddo) 671
  deleteObject(ddo) 672
  evaluate 679, 681
  execute 679, 681
  executeWithCallBack 679, 681
  listAttrs 675
  listEntities 674
  listEntityAttrs(entityName) 675
  moveObject 674
  removeFolderItem(folder, mbr) 672
  retrieveFormOverlay 671
  updateObject(ddo) 671
DKDatastoreTS
  clearIndex 680
  connect 679
  createIndex 680
  deleteIndex 680
  disconnect 679
  evaluate 679
  execute 679
  executeWithCallBack 679
  getIndexFunctionStatus 680
getIndexInformation 680
setIndexFunctionStatus 680
startUpdateIndex 680

DKDDO
   del 672
   retrieve 671
   setData 671
   update 671

DKFolder
   addMember 672
   removeMember 672

DKParts
   addMember(doc, part) 673
   removeMember(doc, part) 673

dkResultSetCursor
   fetchNext for combined search 681
   fetchNext for parametric search 679
   fetchNext for text search 679
   fetchObject for combined search 681
   fetchObject for parametric search 679
   fetchObject for text search 679

DKStorageManageInfoDL class 673

DKUserMgmtDL
   add 676
   del 676
   update 676

DKWorkBasketDL
   getNextHighPriorityItem 678
   listItemIDs 678
   listWorkManagementInfos 678

DKWorkFlowServiceDL
   completeWorkFlowItem 677
   listWorkBasketIDs 678
   listWorkBaskets 678
   listWorkFlowIDs 678
   listWorkFlows 678
   routeWipItem 677
   startWorkFlowItem 677

DL Connector Layer 455

DLL 40
DLLs 83
DMS 8
Document 119
document 30, 44, 48, 58
document collections 23, 553
document item type 50
document management 3
document management parts 50–51
Document Management System 8

Document part 119
document part 30, 44, 58, 72
document parts 48, 50, 71
Document Routing 75–77, 115
document routing 24
Document Routing APIs 148
document routing tables 47, 49
document summarization 6
domain 214
   assigning collection to 218
   assigning privilege set to 217
   assigning Resource Manager to 218
   assigning user group to 217
   assigning user to 217
domain definition 216
domains
   accessing 217
   administering 216
   enabling 215
   enabling consideration 215
Domino 5
double 32
download 25
DRM 421
dsm.opt 253
dsm.opt.smp 252
dsm.sys 253
dsmacct.log 423
DSMI_CONFIG 421
DSMI_DIR 421
DSMI_LOG 421
dsmserv.dsk 422–423
dsmserv.opt 425
DSMSERV_ACCOUNTING_DIR 423
duplicating the resources 60
Dynamic Data Objects 148
dynamic link libraries 83
Dynamic SQL 70

E
EBCDIC 435, 443
EBPP 7, 13
e-business 3
eClient 25, 288
ECM 20
EIP 458
EJB 228
Electronic Bill Presentment and Payment 7, 13
Electronic Statement Presentment 7
enable text search 119
enabling text search 123
encryption 189
encryption key 219
encryption key management 221
enterprise configuration 241
enterprise console 241
Enterprise Content Management 20
Enterprise Information Portal 5
environment variables for C++ 137
environment variables for Java 135
error code
 z/OS 656
error log file 589
escape sequence summary 166
escape sequences 165
  using it in Java and C++ 171
  using it with advanced text search 170
  using it with basic text search 169
eSP 7
events table
  removing entries 483
EXCEPT 180
Exchange Server 7
exit routines 26, 212
export utility 446
Extended Data Object 148

F
failover 20, 297
Federated Connector
  porting 458
federated connector 26
Federated folders 454
Federated Layer 455
file system volume 480, 570
FileNet Panagon imaging services 5
firewall 225
floating point numbers 32
FM 460
folder 45, 58
Folder Manager
  porting 460
folder manager APIs
  mapped to ICM connector
    lp2ActivateItem 678
    Ip2AddGroup 676
    lp2AddUser 676
    lp2AddUserToGroup 676
    lp2CloseTOC for parametric search 679
    lp2CloseTOC for workflow 678
    lp2CloseTOC when retrieving items 671
    lp2CompleteWorkflow 677
    lp2CreateAttr 675
    lp2CreateClass 674–675
    Ip2CreatePrivSet 676
    Ip2DeleteAttr 675
    Ip2DeleteContentClass 677
    Ip2DeleteGroup 676
    Ip2DeleteIndex 675
    Ip2DeletePrivSet 676
    Ip2DeleteUser 676
    Ip2GetNextWorkBasketItem 678
    Ip2ListAttrs 675
    Ip2ListUser 676
    Ip2ListWorkBaskets 678
    Ip2ListWorkFlows 678
    Ip2ModifyAttr 675–676
    Ip2ModifyContentClass2 677
    Ip2ModifyGroup 676
    Ip2ModifyPrivSet 676
    Ip2ModifyUser 676
    Ip2QueryObjectAccess 672
    Ip2RemoveUserFromGroup 676
    Ip2RouteWipItem 677
    Ip2SMSCreateEntity 677
    Ip2SMSDeleteEntity 677
    Ip2StartWorkflow 677
    Ip2SuspendItem 677
    Ip2UpdateAccessList 676
  SimLibAddFolderItem for adding items to
    folder 672
  SimLibAddFolderItem for creating items
    671
  SimLibAddFolderItem for updating items
    671
  SimLibCatalogObject 672
  SimLibChangeIndexClass 674
  SimLibChangeObjectSMS 673
  SimLibCloseAttr 671, 674
  SimLibCloseObject 672
  SimLibCreateItemPartExtSrch for adding
    and indexing parts 680
  SimLibCreateItemPartExtSrch for creating
    items 671

700  Content Manager Implementation and Migration Cookbook
SimLibCreateItemPartExtSrch for creating parts 672
SimLibCreateItemPartExtSrch for importing parts 672
SimLibCreateItemPartExtSrch for updating items 671
SimLibCreateItemPartExtSrch linking objects and items 673
SimLibDeleteItemPartExtSrch for deleting items 672
SimLibDeleteItemPartExtSrch for deleting parts or resource items 672
SimLibDeleteItemPartExtSrch for updating items 671
SimLibDeleteObject for deleting items 672
SimLibDeleteObject for deleting parts or resource items 672
SimLibDeleteObject for removing parts from an item 673
SimLibDeleteObject for updating items 671
SimLibDeleteItemPartExtSrch 673
SimLibGetAffiliatedTOC for combined search 681
SimLibGetAffiliatedTOC for combined searching 681
SimLibGetAffiliatedTOC for parametric search 679
SimLibGetAffiliatedTOC for retrieving items 671
SimLibGetAttrInfo 675
SimLibGetIndexClassView 671
SimLibGetItemAffliatedTOC 672
SimLibGetItemInfo 679
SimLibGetItemInfo for combined search 681
SimLibGetItemSnapshot 671, 679
SimLibGetTOC for combined search 681
SimLibGetTOC for parametric search 679
SimLibGetTOC for retrieving items 671
SimLibGetTOC for workflow 678
SimLibIndexClassView 681
SimLibIndexPartExtSrch 680
SimLibInvokeSearchEngine 680
SimLibInvokeSearchEngine when adding and indexing parts 680
SimLibInvokeSearchEngine when adding parts to an item 673
SimLibInvokeSearchEngine when creating items 671
SimLibInvokeSearchEngine when creating parts 672
SimLibInvokeSearchEngine when updating parts to an item 673
SimLibListClasses 674
SimLibListClassViews 674
SimLibLoadMediaObject 672
SimLibLoadMediaObject when adding and indexing parts 680
SimLibLoadMediaObject when adding parts to an item 673
SimLibLoadMediaObject when creating items 671
SimLibLoadMediaObject when creating parts 672
SimLibLoadMediaObject when updating items 671
SimLibOpenByUniqueName 671
SimLibOpenItemAttr 671, 674
SimLibOpenObject when exporting parts 672
SimLibOpenObject when prefetching parts 673
SimLibOpenObject when retrieving parts 672
SimLibQueryObject 672
SimLibReadObject 671
SimLibRemoveFolderItem 671–672
SimLibSearch 679
SimLibSearch for combined search 681
SimLibSearch for parametric search 679
SimLibSeekObject 671
SimLibSetIndexClassView for combined search 681
SimLibSetIndexClassView for parametric search 679
SimLibSetIndexClassView when retrieving items 671
SimLibStoreNewObject for adding and indexing parts 680
SimLibStoreNewObject for adding parts 673
SimLibStoreNewObject for creating items 671
SimLibStoreNewObject for creating parts 672
SimLibStoreNewObject for importing objects
SimLibStoreNewObject for updating items 671
SimLibStoreObject 680
SimLibStoreObject for adding parts 673
SimLibStoreObject for creating items 671
SimLibStoreObject for creating parts 672
SimLibStoreObject for importing parts 672
SimLibStoreObject for updating items 671
SimLibUpdateObject 671
SimLibUpdatePartExtSrch 671–672
SimLibWriteAttr 671, 674

Folder Manager Layer 455
folders 293
foreign key 66–67
foreign keys 30, 59, 61
free text search 22
frnpwigc 464
frnpwige 464
frnpwigo 464
frnpwigr 464
frnpwigt 464
frnpwisr 464
frnpwips 464
FTP 6, 21, 23

G
global security 224–225, 228, 578
grammar 165

H
halt 424
hardware and software requirements 293
heap sizes 552
hierarchical data models 22
hierarchical item type 36
Hierarchical Storage Management 149
hints and tips
  migration 654
  migration from ImagePlus 657
history 58
host identifiers
  translating from Windows identifiers 642
HSM 149
HTML 5
HTTP 21, 23
HTTPS 570
https 225

I
IBM Directory Server 189
IBM HTTP Administration Server 605
IBM Key Management Utility 605
IBM SecureWay Directory 189
IBM VisualAge C++ Professional Batch compiler 615
ICM APIs 140
ICM connector 132, 144
ICM connector libraries and packages 135
ICM Connectors 456
icmacixt.c 214
ICMACLPrivExit 212
ICMACLXT 212
ICMAnnotations 51
ICMBase 50
ICMBASESTREAM 48
ICMBaseStream 50
ICMBASETEXT 48
ICMBaseText 50
ICMClient.err 589
ICMCONCT 202
ICMDefineSystemItemTypes 645
ICMFORMS 43
ICMGenPrivExit 212
ICMGENXT 212
icmgent.c 214
ICMIMPL 401
icmimpl 399
ICMIMPO 406
ICMMBKUP 233
ICMMCACL 653
ICMMCOMP 234
ICMMDATA 233
ICMMDFUR 234
ICMMLSDR 651
ICMMLSLD 651
ICMLSLPP 639
ICMMMI71 653
ICMMMI72 653, 655–656
ICMMMI73 655
ICMMMI7A 656
ICMMMICx 659
ICMMMIFx 659
ICMMMPx 659
ICMMRACF 232
ICMMRMDR 651
ICMMSORT 234
ICMMUSTB 234
icmn*mak 214
ICMNoteLog 50
ICMPARTS 129, 179
ICMPUBLIC 199
ICMPublic 215
icmrm.logfile 579
ICMRM.properties 202, 205, 253, 425
icmrm_asyncr_logging.xml 492, 580
icmrm_loggin.xml 548
icmrm_logging.xml 580
icmrm_migrator_logging.xml 580
icmrm_purger_logging.xml 580
icmrm_replicator_logging.xml 580
icmrm_stager_logging.xml 580
icmrm_validator_logging.xml 494, 580
ICMRMDEL 491
icmrmlsval.bat 493
icmrmlsval.sh 493
ICMRMTX 491
icmrmvolval.bat 493, 497
icmrmvolval.sh 493, 497
icms*mak 214
ICMSAVEDSEARCH 43
ICMSERVER.LOG 154–155
ICMSTACCESSCODES 207
ICMSTACCESSLISTS 207
ICMSTATTRDEFS 33
ICMSTATTRGROUP 35
ICMSTCOMPDEFS 35
ICMSTCOMPILEDACL 207
ICMSTCOMPILEDPERM 198
ICMSTDOMAINACCESS 207
ICMSTDOMAINPRIVSET 198
ICMSTITEMTYPEDEFS 44
ICMSTLINKSnnnssss 60
ICMSTNLSKEYWORDS 198, 207, 475
ICMSTPRIVDEFS 198
ICMSTPRIVGROUPCODE 198
ICMSTPRIVSETCODES 198
ICMSTPRIVSETS 198
ICMSTResourceMgr 231
ICMSTSYSCONTROL 207, 566
ICMSTSysControl 432, 596
ICMSTUSERGROUPS 199
ICMSTUSERS 199
ICMTRACE_BASIC 596
ICMTRACE_CACHE_ALLOC 596
ICMTRACE_CACHE_MGT 596
ICMTRACE_CACHE_TRACE 596
ICMTRACE_DATA 596
ICMTRACE_DETAILED 596
ICMTRACE_MEMDEBUG 596
ICMTRACE_OFF 596
ICMTRACE_PARSEBUILD 596
ICMTRACE_PERFORMANCE 596
ICMUDBRM 638–639
ICMUJOBS 656
ICMULOAD 638–639
ICMUOBJ 638
ICMUOBJ1 638
ICMUPGM 638
ICMUSQL 638
ICMUTnnnnnssss 58
icmx*mak 214
ICMXLSLG 232
ICSPARM 597
identifiers
   translating from Windows to host 642
IIOP 226
IKEYMAN 605
image annotation 25
Image Search Engine
   porting 461
ImagePlus 658–659
ImagePlus for OS/390
   migrating from 656
import users 191
import utility 234
   Library Server 399–400
index
   reorganizing 128
   updating 127
index class 72
indexes 69
Information Center 160
information extraction 6
information gathering template 286
Information Integrator for Content  5, 133, 458
Information Mining  6
Information Mining based applications
   porting 463
infrastructure 20, 290
installation
   preparation and planning for z/OS 636
installation for z/OS 635
installation verification
   for AIX 623
   for Sun Solaris 635
installing Content Manager on AIX 620
installing Content Manager on Sun Solaris 631
installing Content Manager on Windows 608
integer 32
Internet Inter-ORB Protocol 226
INTERSECT 180
ISO EBNF notation 165
item 30, 44–45, 57, 72, 291
item type 30, 43, 72, 119, 291
item type classification 57
item type subset 55
item type table 47–48
item type views 162
item types 21, 43–44
items 21
ItemSuperAccess 207
ItemTypeClass 44
IXF 435

J
J2EE 228
JAAS 226
Java 2 Connector 226
Java 2 security 228
Java 2 Security Manager 226
Java Authentication and Authorization Service 226
Java beans
porting 457
Java code samples 143
Java Server Pages 26
Java Viewer toolkit 141
Java Virtual Machine 552
java_heap_sz 552
JavaBeans 26, 115
JavaServer pages 228
JCL 233–234, 637, 639, 653
jobs
run jobs in parallel 655
JPEG 39
JPEG images 39
JSP 26
JSP Taglib 141
JSPs 228
JVM 548, 552

K
key field 72
keyword item 45
keyword list 45
KEYWORDCLASS 198, 207
knowledge management 3
Kofax Ascent Capture and Captive/Input Accel 12–13

L
LAN Cache 553
LAN cache 23, 294, 297, 480
language identification 6
LBOSDATA 359, 365, 478, 502–503
monitoring directory size 477
LDAP 5, 23–24, 188, 224–225
LDAP authentication 189
LDAP integration 190
for AIX 623
for Sun Solaris 635
for Windows 611
LDAP User ID Import Scheduler utility 191
libraries 639
SICM* 637
Library Client API
porting 460
Library Server 20–21, 25, 33, 132, 188, 288, 550
capacity planning 294
configuration for AIX 621
configuration for Sun Solaris 633
configuration for Windows 608
database tables related to ACL 207
database tables related to privileges 198
database tables related to users 199
import user data 395
import utility 399–400
installation for z/OS 639
logging and tracing 154
merging 429, 444
pinpoint problem 564–565
trace level 566, 596
tuning 552
Library Server database 359
establish connection 371
Library Server ID 436
Library Server log 596
light weight third-party authentication 192
Lightweight Directory Access Protocol 188, 225
LIKE operator  168
line data  6
link  60, 73
links  21, 30, 59, 61, 71, 176
load balancing  20
LOB  41
localhost restriction  606, 618, 629
locking  152
log files
  removing  484
Log4J  579
logbufsz  552
logfilesz  552
logging  154
logheap  552
logon in z/OS  232
logprimary  552
logsecond  552
long integer  32
long-term storage  237
Lotus Discovery Server  8
Lotus Domino  7
Lotus Domino Directory Notes Address Book  189
Lotus Domino.Doc  5
Lotus Extended Search  5
Lotus Notes Domino databases  5
Lotus WordPro  39
ls.ddl  433
LTPA  192

M
MAC  219
magnetic disk  23
maintenance  24, 472
  review  507
  z/OS  507
manage object migrations
  TSM, advantage  250
management class  244, 247
maxagents  552
maxfilop  552
MaxUserPort  551
media  248
Media Object  46, 49
Media Object Class  47, 119
media object class  30, 40
media object class table  47, 49
media object classes  42
memory  545
memory foot print  20
merging Library Servers  429, 444
message authentication code  219
meta data  21
meta model  71
metadata  30, 46
Microsoft Active Directory  189
Microsoft Exchange  5
Microsoft Visual Studio .NET  454
Microsoft Word  39
mid-tier  288
migrate
  Content Manager systems earlier than V7  360
  migrate install on AIX  424
migrating
  within Windows platform  370
migrating Resource Manager on same machine  403
migration  23
application  451
between AIX and z/OS  429
cross-platform  427, 442
from CM z/OS to MP  430
from third-party product  445
hints and tips  654
job flow, parallel  655
on z/OS  649
post  425
TSM  418
migration consideration  432
migration considerations  357
migration flow
  general, data  431
migration hints and tips
  ImagePlus  656
migration jobs
  preparing for z/OS  652
  running on z/OS  653
migration paths  428
migration policies  23
migration prerequisites
  for z/OS  651
migration process  358
migration scenario  448
migration tasks
  for z/OS  650
migration to TSM on AIX  421
migration to TSM on Windows  419
migration validation
   for z/OS 654
Migration Wizard 381, 398–399
migration.log 397, 402
MIME 30
MIME type 39
model 30
modeling entities 70
mon_heap_sz 552
monitoring 549
   performance 546
monitoring AIX 547
monitoring Content Manager 548
monitoring DB2 547
monitoring WebSphere Application Server 548
monitoring Windows 546
moveObject() 154
MP3 audio 39
multi-level child components 38
multiple item types 174
multi-valued attributes 33, 72

N
NAB 189
native connectors 26
Net Search Extender 118, 164
network 545
NoAccessACL 211
node 251
node name 239
NOINDEX 43
Non-visual JavaBeans 141
note 59
NSE 118, 164
NtfsDisable8dot3NameCreation 549
numdb 552

O
OAM 152, 506
object 39–40, 72, 188, 291
Object Access Method 152
Object Server 361, 368, 370, 398, 418, 456, 651
   clearing staging area 365
Object Server database 359
object storage file-system
   create for Resource Manager 620, 631
object storage repository 255
object-oriented interface 141
objects 21, 23, 39, 45
   accessing 218
   objsrvrn 398
OBJSTAT 658
OCR 48
ODBC 377
ODMA 25
OLE 142
OLE API based applications 463
OnDemand 4–7
OO API 129
Open Document Management API 25
operator 168
operators 167, 180
operators for parametric search 163
optical 250
Optical Character Recognition 48
optical disks 448
optimizer 70
options checklist
   planning and designing 299
Oracle 21, 198
orphan 496
overflow storage area 255
overflow storage system 255
overflow volumes 255
overload limit 84, 89

P
packages 464
parametric search 22, 163
parent component 37, 72
Parent ID 37
part 72
partial keyword search 22
parts table 49
password 202, 230
passwords 200
   changing 205
PDFs 6
performance
   planning 295
   text indexing 130
Performance Expert 553
performance goals 544
performance monitoring 546
Performance Monitoring Infrastructure 548
performance requirements 290
Index

[463x26]Index

707

performance tuning 545
  basics 544
  guideline 545
PID 148
plain text files 39
planning 285
  capacity 294
  trade-offs 295
planning for performance 295
PMI 548
policy 240
policy domain 244
policy objects 242
policy set 244
port 606
port 9090 574
porting
  Client for Windows Automation Interface 462
  DL Connector 459
  Federated Connectors 459
  Folder Manager 460
  Information Mining based applications 463
  Java beans 458
  Library Client API 460
  QBIC 461
  TSE 460
porting process 452
ports
  restricting 606, 618
pre-configured ACLs 211
pre-defined privilege sets 197
pre-defined privileges 196
prerequisite verification
  for AIX 613
  for Sun Solaris 624
  for Windows 602
print output 6
privilege 197
  ItemSuperAccess 207
  UserDB2Connect 203
privilege access control 212
privilege group 197
privilege set 196, 199
  assign to ACL 211
privilege sets 77, 195, 197, 206, 293
privileges 195–196, 206
  Library Server database tables 198
problem 559
  pinpoint 564
process 79, 103, 293
processes 78
PROCLIB 638–639
profile.env 617
programming options 141, 143
programming tips 464
property 31
prototype system 289
PTFs 650
PubAccessEnabled 207
public Access 208
PublicDomain 216
PublicReadACL 211
purge 365
purge rates
  alter to completely clear staging area 368
purger 367, 481

Q
QBIC
  porting 461
queries 70
query 21
  dealing with lists 179
query against attributes
  resource items 178
  system attributes 178
query against document parts 179
query APIs 146
query flow 163
query language 22, 161–162, 165, 172
query options 182–183
query results 182, 184
query string 172–173, 182

R
RACF 187, 229–230
RACF batch import utility 233
RACF exit program 234
RACF import utility 233
RACF user exit 230, 232
RACFBKUP 233
RACFDATA 233–234
RACFSORT.OUT 234
rapid application development toolkit 26
RDBMS 21, 185
README_SAMPLES_C++_ICM.txt. 140
README_SAMPLES_JAVA_ICM.txt 140
rebind 70, 551
recovery 237
recovery log 238
Redbooks Web site 693
Contact us xx
reference 65–66, 73
references 21, 30, 59, 61, 71, 177
referential integrity 436
referential integrity constructs 433
REG_DWORD 551
regedit 549
regenerate Web server plugin 574
relational database management system 21
Relational Database Manager System 185
relationships 30
Remote Method Invocation 226, 464
REORG 475
reorganize index 128
REORGCHK 474
reorgTextIndexes 129
repartitioning hard disk 499
replacement tokens 641
replication 20, 23, 297, 553
report template 6
reports 496
validation discrepancy 498
repositories 24, 26
repository 23
requirements 286, 290
hardware and software 293
RESCOMP.OUT 234
resource item 30, 44, 58, 71–72, 119, 178
resource items 46
Resource Manager 23, 33, 132, 288, 370, 550
capacity planning 294
communication in z/OS 232
configuration for AIX 621
configuration for Sun Solaris 633
configuration for Windows 609
configuring to use TSM 253
create object storage file-system 620, 631
create staging directory 620, 631
database on z/OS 641
installation for z/OS 641
logging 156, 579
logging appender-ref values 581
logging control files 580
logging priority values 580
migrating on different machine 407
migrating on the same machine 403
migrating user data 403
migrator log 593
pinpoint problem 565, 570
recreate database on AIX 571
recreate database on Solaris 572
trace 597
tuning 553
utilities and services 485
Resource Manager database 501
Resource Manager objects 150
Resource Manager volume validation utility 485, 497
Resource Manager/Library Server utility 485
Resource Manager/Library Server validation utility 494
Resource Managers 20, 214
resource services
starting and stopping 490
response time 25, 544
restore 237
restricting ports 618, 629
retrieve 237
retrieveObject() 153
RI s 436
RMI 226, 464
RMLSITEMS 495
RMMigrator 485
RMPurger 485
RMReplicator 485
RMStager 485
RMSTAGING 501
RMVOLUMES 502
rnpwigi 464
rollback 151
rollback() 153
root 35
root component 30, 36, 43, 60, 65, 71
RUNSTATS 475
runstats 70, 551
S
sample code 139, 160, 185, 232, 467
SAP 7
SAP ArchiveLink 8
SAS 226
SDocRoutingDefinitionCreationICM 115
SDocRoutingDefinitionDeletionICM 115
SDocRoutingListingICM  115
SDocRoutingProcessingICM  115
search  22, 182
search indexes  5
search options  123
search string symbols  165
search using the latest-version  178
searching  117
searching for documents  129
searching for object contents  129
Secure Sockets Layer  188, 192, 228
configuring for IBM HTTP Server  605, 617, 628
Secured Sockets Layer  583
security  7, 20, 187–188, 206
planning and designing  298
z/OS  229
Security Authentication Service  226
security domain  225
security model  188
security token  188, 220
security token generation  219
security token validation  220
security tokens  219
SegmentID  58, 60
semantic type  30, 58–59
separation policies  550
server databases
optimizing  472
server exits  112
server storage  237
servers
shut down  369
service level agreements  290
Servlet  141
set up
WebSphere Studio Application Developer  137
set up users
for AIX  615
setting up environment
application development  135
shared object  40
short integer  32
shutting down servers  369
SICM* libraries  637
SICMCPY1  637
SICMDBR1  637
SICMDBR2  638
SICMINS1  637, 651, 653
SICMINS2  638
SICMLMD1  637
SICMLMD2  638
SICMLMM1  638
SICMMDB1  638
SICMMIN1  638
SICMSAM1  232, 638
single Sign-on  579
single sign-on  23, 189, 192
  enabling through database authentication  195
  enabling through WebSphere security  192
SISMINS1  233
SItemUpdateICM  154
size mismatch  496
SMP/E  637, 651
SMS collections  214
SMS server service  368
software prerequisites
for AIX  612
for Sun Solaris  624
Windows  602
SORTBY  162
SQL queries  22, 185
SSearchCallbackObjectICM  185
SSearchICM  185
SSL  188, 192, 228, 570, 583, 605
  configuring for IBM HTTP Server  617, 628
staging area  188, 482–483
  alter purge rate to completely clear it  368
  clearing  365
  managing directory space  480
staging area purge process
settings  367
staging directory  23
  create for Resource Manager  620, 631
staging volume
  replacing  501
  replacing for AIX and Solaris  499
  replacing for Windows  501
Start Here CD  290
startTransaction()  152–153
stmtheap  552
storage  237
storage area
  overflow  255
storage class  553
storage classes  23
storage hierarchy  247
storage management  24, 249
storage management applications  238
storage policy 247
storage pool 245, 247
storage pools 248
storage volume
   replacing for AIX and Solaris 500
   replacing for Windows 502
stored procedures 21
streaming video 5
streaming video and audio 20, 24
super administrators 201
SuperDomain 216
symbolic link type 61
symbols used within search string 164–165
SYSINDEXES 473, 475
SYSPRINT 596
SYSPTINT 597
system administration
   configuration for Windows 610
System Administration Client 24
system configuration 290
System Flag 436
system management APIs 147
system topology
   planning and designing 288
system-defined attributes 65
system-defined privileges 196

T
table joins 38
Taglib 141
tape 250
tape rotation 248
TAPECLASS 423
taxonomies 6, 21
taxonomy 20
TCA 253
TcpTimedWaitDelay 551
template 43
   analyze business operations 286
text index 129–130
text indexing 117
text search 118, 129, 163–164, 169–170
   planning and designing 298
Text Search Engine
   porting 460
text search on attributes
   within an attribute group 180
text search options 124
the TSM client API options file 253
threshold cycle 483
throughput 544
TIE 22
TIREF 129
TIFF 39
time 33
time stamp 33
Tivoli Performance Viewer 548
Tivoli Storage Manager 7, 235, 249, 418–419, 478
   architecture 238
   overview 236
Tivoli Storage Manager API client 253
tivoli.tsm.license 425
TMV 548
token 188, 565, 570
   duration 222
   validation 220
tokens 219
topology 290
trace 559
   z/OS 595
TRACELEVEL
   Library Server 566
tracing 154
trade-offs 545
   capacity planning 295
transaction 151–153
transaction reconciliation utility 491
transaction scope 151
transactions 150
triangular architecture 20–22
troubleshooting 20, 564
TSE
   porting 460
   administration 240
   architecture 238
   configuring Resource Manager to use it 253
   manage object migrations advantages 250
migration 418
migration on AIX 421
migration on Windows 419
   overview 236
TSM API 253
TSM API client files
   customizing 252
TSM archive operation 242
TSM Backup Archive Client 252, 594
TSM backup/archive client 239
TSM client API 251
TSM Client API log 594
TSM client interface 237
TSM client nodes 244
TSM integration
troubleshooting 593
TSM levels
supported by Content Manager 251
TSM license 249
TSM management class 255
TSM media and device policy objects 245
TSM policy objects 242
TSM policy objects in action 243
TSM Server 595
TSM server 238, 251–252
TSM server configuration 252
TSM server tasks 241
TSM server versions 251
TSM storage management environment 247
TSM storage pool 254
TSM storage pools 251
TSM volume 254
tuning AIX 550
tuning Content Manager 552
tuning DB2 550
tuning WebSphere Application Server 551
tuning Windows 549
Two-tier configuration 288

U
UDF declaration for ACL exits 212
udf_mem_sz 552
Uniform Resource Identifier 150
Uniform Resource Locators 150
UNION 180
update index 127
updateObject() 153
URI 150
URL 150
user
privileges 206
user access rights
sample calculation 196
user data
import into Library Server 395
migrating into Resource Manager 403
user exit routines
access control 212
user group 199
user groups 195
user ID 202
creating 200
installation for z/OS 229
User ID Import Scheduler utility 191
user IDs
create 626
create for AIX 615
give privileges 616
user privileges 196
UserDB2Connect privilege 203
user-defined attributes 36
user-defined privileges 196
user-exits 142
USERKIND 199
userinfo 617, 628
users
assigning to collections 199
give privileges 626
Library Server database tables 199
set up 625
set up for AIX 615
set up for Windows 604
users
assigning to Resource Managers 199
USERTABL.OUT 234

V
vacpp.lic file set 615
validation discrepancy reports 498
validation utilities 493
validation utilities log 494
validation utility parameters 495
variable character 32
version policy 53–55
versioning 20, 22, 52, 73
versions 30, 178
VideoCharger 24
Visual Beans 141
VisualAge C++ 615
VOL_LOGICALNAME 502
VOL_MOUNTPOINT 502
volume
TSM policy object 245
volumes
overflow 255

W
Web Crawler 6
Web crawling 6
Web search engines 5
Web server plugin
  generating 584
  regenerate 574
Web-based Systems Manager Server 575
WebSphere Administrative Console 223
WebSphere Application Server 223
  problem starting 574
WebSphere global security 228
WebSphere MQ Workflow process 5
WebSphere Process Server (WPS) 76–77
WebSphere security 192
WebSphere SSL configuration 606, 618, 629
WebSphere Studio Application Developer 137
Windows client 25, 588
Windows client log file location 589
Windows identifiers
  translating to host identifiers 642
WLM 596
work baskets 77, 83
work node 78, 112, 293
work package 78, 83, 112, 115
work packages 78, 83
workflow 20, 24–25, 76–77, 454
  key concepts 292
  planning and designing 292
workflow branching 80
worklist 103, 106
workstation
  set up 642
workstation authentication 192
workstation log 597
WORM media 250
WSM 575

X
XBSA 242
XDO 42, 46, 49, 465
XDOs 148, 150
Xerox 6
XML 5, 162, 446
XML import capabilities 454
XMLPath 22
Content Manager Implementation and Migration Cookbook
This IBM Redbook deals with IBM DB2 Content Manager implementation and migration. It is aimed at architects, designers, developers, and system administrators of Content Manager systems.

We cover the basic concepts needed to design and implement a Content Manager solution. This includes topics on data modeling, workflow, text indexing and search, application development, query language, security, and Tivoli Storage Manager (TSM) overview. In addition, we cover the solution implementation process from planning and designing, to deployment. To put concepts into real practice, we provide practical case studies to demonstrate how to implement a Content Manager solution for real-world scenarios.

We also deal with IBM DB2 Content Manager migration. This includes migration on multiproducts, for TSM, and for Content Manager custom applications. In addition, we describe an approach and process for special migration scenarios such as cross platform migration and migration from a third-party product.

Once a Content Manager system is implemented or migrated, it is important to maintain the system. We discuss maintenance, performance tuning, and troubleshooting issues for a production Content Manager system.