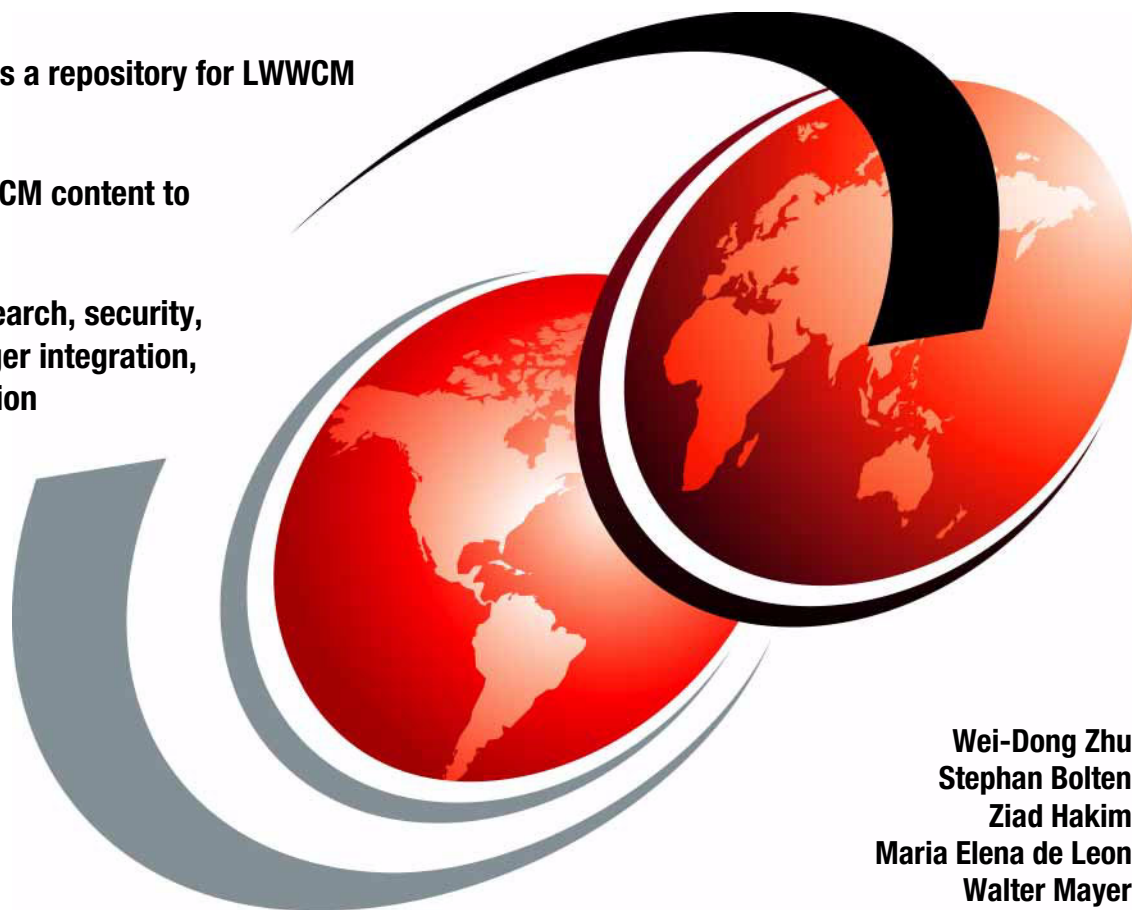


Lotus Workplace Web Content Management and Content Manager Working Together for LWWCM Java Edition V2 and CM V8 Only

Using CM as a repository for LWWCM

Publishing CM content to
LWWCM

Covering search, security,
VideoCharger integration,
and migration



**Wei-Dong Zhu
Stephan Bolten
Ziad Hakim
Maria Elena de Leon
Walter Mayer**



International Technical Support Organization

**Lotus Workplace Web Content Management and
Content Manager Working Together**

September 2004

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

First Edition (September 2004)

This edition applies to Version 2 Release 1 of Lotus Workplace Web Content Management Java Edition (product number 5724-I29), Version 8, Release 2 of IBM DB2 Content Manager for Multiplatforms, IBM DB2 Content Manager Information Integrator for Content for Multiplatforms (product numbers 5724-B19, 5724-B43).

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

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

Contents

Figures	ix
Notices	xiii
Trademarks	xiv
Preface	xv
The team that wrote this redbook	xv
Become a published author	xvii
Comments welcome	xviii
Part 1. Introduction	1
Chapter 1. Lotus Workplace Web Content Management overview	3
1.1 Introduction	4
1.2 LWWCM architecture	5
1.2.1 LWWCM user interface	7
1.3 Creating and updating Web sites	9
1.3.1 Information Architecture (IA)	9
1.3.2 Web page design	9
1.3.3 Collaboration for content management and distribution	10
1.3.4 Versioning	11
1.3.5 Syndication	12
1.4 Retrieving data from different sources	12
Chapter 2. Content Manager overview	15
2.1 Why implement an Enterprise Content Management infrastructure?	16
2.2 Content Manager product portfolio components	17
2.2.1 IBM Lotus Workplace Web Content Management (LWWCM)	17
2.2.2 IBM DB2 Document Manager	18
2.2.3 IBM DB2 Records Manager	18
2.2.4 IBM DB2 Content Manager Information Integrator for Content.	18
2.2.5 IBM DB2 Content Manager OnDemand	20
2.2.6 IBM VideoCharger (bundled with Content Manager)	20
2.2.7 IBM Tivoli Storage Manager	20
2.2.8 IBM DB2 CommonStore	20
2.2.9 IBM Electronic Media Management System (EMMS)	21
2.2.10 IBM DB2 UDB Net Search Extender (NSE)	21
2.3 Content Manager architecture overview	22
2.3.1 Library Server	23

2.3.2	Resource Manager	26
2.3.3	System Administration Client	29
2.3.4	Content Manager clients	30
2.3.5	How they work together	31
2.3.6	Working with LWWCM	32
Chapter 3.	Why and when to integrate	35
3.1	Reasons to integrate	36
3.2	Benefits of integration	36
3.2.1	Integrating using Content Manager as the repository	37
3.2.2	Integration of external resources	37
3.3	Value propositions	38
3.3.1	LWWCM	38
3.3.2	Content Manager	39
3.3.3	Integrated solution	39
3.4	Advantages of CM compared to other repositories	39
3.5	Sample scenario	40
Chapter 4.	Case study	41
4.1	Introduction	42
4.2	Adding the insurance option into the navigator	43
4.2.1	Creating new options under the insurance option	46
4.3	Creating content for the new insurance options	48
4.3.1	Insurance workflow	48
4.3.2	Content templates	49
4.3.3	Insurance page design	53
4.3.4	Creating insurance content	56
4.4	Retrieving data from external sources into insurance content	58
4.5	Troubleshooting	59
Part 2.	Integration	61
Chapter 5.	Before you begin integration: pre-integration validation	63
5.1	Introduction	64
5.1.1	Important log files and configuration files	64
5.1.2	Some network considerations	65
5.2	Software requirements and case study setup	67
5.2.1	Software requirements for LWWCM	67
5.2.2	Software requirements for Content Manager	68
5.2.3	Case study setup	70
5.2.4	How to check installed software versions	71
5.3	Validation checklist	72
5.3.1	WebSphere Application Server and IBM HTTP Server	72
5.3.2	LWWCM	73

5.3.3	DB2 and NSE	74
5.3.4	IBM Content Manager and Information Integration for Content	75
5.4	Problem determination and troubleshooting	78
5.4.1	General hints and tips	78
5.4.2	WebSphere Application Server	79
5.4.3	DB2 and NSE	80
5.4.4	IBM Content Manager and Information Integration for Content	83
5.4.5	LWWCM	87
Chapter 6. Using CM as a repository for LWWCM		91
6.1	Configuring LWWCM to use CM as a data repository	92
6.1.1	Prerequisites	92
6.1.2	Modifying the WebSphere Application Server JVM setting	92
6.1.3	Modifying LWWCM's aptrixjpe.properties file	94
6.1.4	Deleting existing indexes from the LWWCM server	96
6.1.5	Starting LWWCM and running the initial setup	96
6.1.6	Troubleshooting	96
6.2	Accessing a remote Library Server database	97
6.3	LWWCM server routing information to access CM	99
6.3.1	The aptrixjpe.properties file	101
6.3.2	Files cmbicmsrvs.ini and cmbds.ini	103
6.3.3	Testing the ICM and FED connectors	107
6.4	How LWWCM objects are stored in CM	108
6.4.1	Viewing AJPEData and AJPEResources item types	109
6.4.2	Storing LWWCM objects in AJPEData/AJPEResources objects ..	109
6.4.3	Viewing AJPEData and AJPEResources items	110
6.4.4	Viewing AJPEData XMLContent data	113
Chapter 7. Publishing Content Manager content		115
7.1	Introduction	116
7.2	Prerequisites	117
7.2.1	LWWCM server	117
7.2.2	Content Manager server	118
7.2.3	Content Manager data	118
7.3	Federated content component	123
7.3.1	What is a federated content component?	123
7.3.2	Why use federated content components?	124
7.3.3	Setting up a federated content component	124
7.3.4	Putting it all together	136
7.4	Connect tags	139
7.4.1	What are connect tags?	139
7.4.2	Setting up connect tags	139
7.4.3	Putting it all together	148

7.5 JSP component	151
7.5.1 What is a JSP component?	151
7.5.2 Why use the JSP component?	152
7.5.3 Setting up a new JSP component	152
7.5.4 Sample code	156
7.5.5 Putting it all together	162
7.6 Troubleshooting	164
7.6.1 Log files and trace levels	164
7.6.2 Basic connection testing	165
7.6.3 Executing the JSP program outside of LWWCM	165
Chapter 8. System architecture considerations	167
8.1 Introduction	168
8.1.1 Using Content Manager as the repository for LWWCM objects . . .	169
8.1.2 Using Content Manager as the repository for external resources . .	169
8.1.3 Server operating system considerations	169
8.2 LWWCM and Content Manager on the same machine	169
8.2.1 LWWCM considerations	170
8.2.2 Content Manager considerations	171
8.2.3 Security considerations	171
8.2.4 Maintenance	171
8.2.5 Advantages and disadvantages	171
8.3 LWWCM and CM on different machines	172
8.3.1 LWWCM considerations	172
8.3.2 Content Manager considerations	173
8.3.3 Security considerations	173
8.3.4 Maintenance	174
8.3.5 Advantages and disadvantages	174
8.4 Integration scenarios	174
8.4.1 One LWWCM server	174
8.4.2 Several LWWCM servers	177
8.4.3 Using Content Manager as a repository for external resources . .	185
8.5 Mixed operating systems (Windows and AIX)	185
Chapter 9. Portal integration	187
9.1 Architecture	188
9.1.1 Content creation	188
9.1.2 Content storage	188
9.1.3 Content management	188
9.1.4 Content delivery	189
9.2 Why integrate LWWCM in WebSphere Portal Server?	190
9.3 Levels of integration	191
9.4 LWWCM portlets	193

9.4.1 Typical LWWCM portal page	193
9.4.2 User management and security	200
9.5 Content Manager portlets	201
9.5.1 Content Manager/Information Integrator for Content portlets	202
9.5.2 Information Integrator for Content portlets	204
Part 3. Advanced topics	207
Chapter 10. Search	209
10.1 Overview	210
10.1.1 Content Manager search possibilities	210
10.1.2 NSE integration within Content Manager	212
10.1.3 Federated search	215
10.1.4 LWWCM search module	219
10.2 Integrating a search.	219
10.2.1 Leveraging the CM/II for Content Java APIs.	220
10.2.2 IFrame.	231
10.2.3 LWWCM internal search	232
10.3 Other search approaches	234
10.3.1 Portal search engine (PSE).	234
10.3.2 Domino Extended Search (DES)	234
Chapter 11. Security	237
11.1 LWWCM security.	238
11.1.1 Access levels and security sections	238
11.1.2 Default users and groups	240
11.1.3 How it works	242
11.2 Content Manager security	248
11.2.1 Authorization	249
11.2.2 Authentication	249
11.2.3 How it works	250
11.3 How security types work together	251
11.3.1 Security in general	251
11.3.2 Security with connect tags	252
11.3.3 Security with federated content component	252
11.3.4 Security with the JSP component	253
Chapter 12. VideoCharger integration.	255
12.1 VideoCharger and how it works	256
12.2 Integration of VideoCharger and Content Manager	259
12.2.1 Configuring a Resource Manager storage managed system	260
12.2.2 Creating a video insurance item type	261
12.2.3 Importing, searching for and playing videos	262
12.3 Integrating Content Manager videos to a Web site	267

12.3.1	Why integrate Content Manager videos to a Web site?	267
12.3.2	Prerequisites	267
12.3.3	Creating video components in LWWCM	268
12.3.4	Incorporating video within a Web page	269
Chapter 13.	Data migration through syndication	271
13.1	Syndication overview	272
13.1.1	Setting up the syndication tool	272
13.1.2	Using unique Content Manager repository	275
13.1.3	Using different repositories	277
13.2	Migrating content repositories	278
13.2.1	Troubleshooting	286
13.3	Upgrading LWWCM V1.1 content to LWWCM V2.0	287
13.3.1	File system to file system upgrade	287
13.3.2	File system to Content Manager migration	288
13.3.3	Enabling LWWCM version control	289
Chapter 14.	Versioning	293
14.1	Versioning overview	294
14.2	Versioning of repository objects	294
14.2.1	Versioning in LWWCM	295
14.2.2	Versioning in Content Manager	302
14.3	Versioning external resources	303
14.3.1	Versioning in federated, JSP components and connect tags	304
14.3.2	Versioning using Content Manager	304
Appendix A.	Installation supplement	309
	Accessing a remote LS database	310
	Cleaning up a Windows system prior to installation	311
Related publications	313
	IBM Redbooks	313
	Other publications	313
	Online resources	313
	How to get IBM Redbooks	314
	Help from IBM	314
Index	315

Figures

1-1	Authors and Web page designers working separately.	5
1-2	LWWCM architecture.	6
1-3	LWWCM user interface with left-side navigator.	8
1-4	Page design and components for a Web page	9
1-5	Workflow stages sample	11
1-6	LWWCM architecture sample	13
2-1	Content Manager product portfolio	17
2-2	Information Integrator for Content architecture	19
2-3	Content Manager architecture overview	23
2-4	Library Server	24
2-5	Resource Manager	27
2-6	Resource Manager manages the storage of the objects.	28
2-7	System Administration Client interface	30
2-8	Clients access to Library Server and Resource Manager	31
2-9	LWWCM and Content Manager working together	33
4-1	IT Jumpstart original main menu	43
4-2	New site area.	44
4-3	Site area properties	45
4-4	Navigator with insurance products option	45
4-5	Insurance parent site area	46
4-6	Insurance sub-menu	47
4-7	Group plans sub-menus.	47
4-8	Save workflow	48
4-9	Workflow stages update.	49
4-10	New workflow content template	50
4-11	Component Manager	51
4-12	Add component in Component Manager	51
4-13	Result of after adding all components in insurance template	52
4-14	First part of the content template components.	52
4-15	Insurance Web page with components added in insurance template . .	53
4-16	Site area properties setting	56
4-17	Life insurance Web page with navigator in the left hand side	58
5-1	WebSphere Application Server administration console.	73
5-2	Zone Labs Integrity desktop.	83
6-1	DB2 Control Center	99
6-2	LWWCM configuration to use Content Manager as repository	101
6-3	II for Content ICM and FED connector installation.	103
6-4	Server configuration utility	106

6-5	AJPEData and AJPEResources item type attributes	108
6-6	AJPEResource document management definition	109
6-7	How LWWCM objects are stored in Content Manager	110
6-8	Content Manager AJPEData items after LWWCM-Content Manager setup 111	
6-9	Windows client AJPEResources search result	111
6-10	LWWCM component library. image component	112
7-1	Installation II for Content: development workstation	117
7-2	Installation Information Integrator for Content: local connectors	118
7-3	New attribute	120
7-4	New Item Type window	121
7-5	Item Type attributes	121
7-6	Item Type Document Management window	122
7-7	Define new resource back-end server: Content Manager V8 server . .	126
7-8	Test connection to the newly defined content server	126
7-9	Refresh server inventory for the newly defined content server	127
7-10	Mapped attributes for the insurance case study	128
7-11	Query with the Insurance Type search template	130
7-12	Query results	130
7-13	Federated Component form	131
7-14	Federated component added within a LWWCM content page	132
7-15	Resulting Web page of publishing CM content using federated components 138	
7-16	Process connect tags	140
7-17	HTML option in Rich Text component	140
7-18	IBM developerWorks Web site	145
7-19	Connect tag to strip partial Web page”	146
7-20	Result of stripping partial Web page content to Insurance Web site . .	146
7-21	Create a new content with connect tag	150
7-22	Resulting Web page of publishing CM content using connect tags . .	151
7-23	JSP component - CM - integration approach	152
7-24	JSP component configuration	155
7-25	Output of CMContentList.jsp	160
7-26	WCM tag sample page	162
7-27	Item Type and Attributes	163
7-28	Setting up new site area	164
8-1	Necessary component for one and many boxes	168
8-2	LWWCM and Content Manager on the same box	170
8-3	LWWCM and Content Manager on different boxes	172
8-4	Secure architecture example: LWWCM/CM in separate boxes	173
8-5	Integration details	175
8-6	Configuration of the AJPEResources to point to another RM	177
8-7	Multi-environment, multi-machines setup	179

8-8	Multiple LWWCM servers/machines scenario architecture	180
8-9	Different LWWCM using the same Content Manager	181
8-10	Several LWWCM servers using the same Content Manager	182
8-11	Multi-LWWCM using multi-Resource Managers	184
8-12	Multi-LWWCM to multi-Content Manager via syndication	185
9-1	WPS/LWWCM / Content Manager architecture	188
9-2	Sample LWWCM portal page layout	193
9-3	Insurance navigator definition	194
9-4	Insurance navigator portlet configuration - content	195
9-5	Insurance navigator portlet configuration - links	195
9-6	Importing text document into Content Manager	196
9-7	Component manager for content Insurance Name 1	196
9-8	Selecting content for the federated content component	197
9-9	Insurance content portlet configuration - content	198
9-10	Insurance content portlet configuration - links	198
9-11	Insurance menu portlet configuration - content	199
9-12	Portal page layout	199
9-13	Insurance portal page	200
9-14	User credentials flow	201
9-15	Content Manager portlet start page	202
9-16	Content Manager portlet items types	203
9-17	Content Manager portlet search entry	203
9-18	Content Manager portlet search result	204
9-19	Federated search portlet - query screen	205
9-20	Federated search portlet - result screen	205
10-1	Parametric search on the InsuranceType attribute IT_Name	211
10-2	Parametric search results	211
10-3	Text search option settings	213
10-4	Import a text searchable document form Content Manager eClient	215
10-5	LWWCM Web site development with federated search objects	216
10-6	JSP with federated search APIs integrated in a Web site	217
10-7	NSE / federated search integration	221
10-8	The content object including the JSP component	228
10-9	Site area properties	229
10-10	NSE and search result Web page	230
10-11	Federated search and search result Web page	230
10-12	eClient integrated through an iFrame	232
11-1	LWWCM object security setting	238
11-2	Workflow buttons	240
11-3	Default author for content created using this template	243
11-4	Add group1 so that the members of this group can see this site	244
11-5	Authors field inherit from Content Template's Author field	245
11-6	Security is from the workflow stage and the author is added	246

11-7	Workflow Security setting in stage1	247
11-8	Workflow security setting for content test 1 after moveing to stage2.	248
11-9	ACL security process flow chart.	251
11-10	Federated content component setting has the first priority	253
12-1	VideoCharger installation.	257
12-2	VideoCharger configuration and administration.	258
12-3	View and run the sample IBM video	259
12-4	VideoCharger storage managed system configuration	260
12-5	Set up VideoCharger SMS configuration.	261
12-6	Selected attributes for the InsuranceVideo Item type	262
12-7	Importing some videos from Content Manager Windows Client	263
12-8	Error while importing a video if IBM LUM is not installed.	264
12-9	Running a basic search against the InsuranceVideo item type	264
12-10	Launching the VideoCharger player application	265
12-11	Open an InsuranceVideo item to play the video	266
12-12	Federated content component for video search	268
13-1	Crossed information between Syndicator and Subscriber.	274
13-2	Content Manager AJPEData table.	276
13-3	Subscriber enable option	276
13-4	Syndication using the one Content Manager repository	277
13-5	Syndication using two different repositories.	278
13-6	Migration approach	279
13-7	Application assembly for temporary LWWCM server.	282
13-8	Syndicator and subscriber for migration	284
13-9	AJPEData search result in Content Manager after syndication.	284
13-10	Deleting the temporary LWWCM application.	286
13-11	Updating the LWWCM Web application.	288
13-12	Migrated content before and after version enablement	291
14-1	Different LWWCM versions in Content Manager	297
14-2	Versioning	297
14-3	Save version of a content object	299
14-4	Version all process status	300
14-5	Version management in LWWCM	301
14-6	Comparison of LWWCM and Content Manager versioning.	302
14-7	Set the versioning policy for the InsuranceVideo item type	306
14-8	Supplemental version policy for specific document parts	307

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:

@server®
e-server®
Redbooks (logo) ™
developerWorks®
ibm.com®
iSeries™
AIX®
Cloudscape™
Domino.Doc®

Domino®
DB2 Universal Database™
DB2®
Everyplace®
Informix®
IBM®
Lotus®
Notes®
PAL®

Redbooks™
Redbooks (logo)™
SLC™
Tivoli®
VideoCharger™
VisualAge®
WebSphere®

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Preface

In this IBM® Redbook, we explore various ways of integrating IBM Lotus® Workplace Web Content Management (LWWCM) V2 and IBM DB2® Content Manager (Content Manager) V8.

In Part 1 of the redbook, we provide brief overviews of LWWCM and Content Manager. We discuss why we might want to integrate both products and when to do so. We also introduce the case study of an insurance company, which we will be using throughout this redbook to show sample integration steps, procedures, and coding.

In Part 2, we cover various areas that you need to be familiar with, aware of, and have set up to achieve a successful integration. We show you how to use Content Manager as a repository for LWWCM, and how to publish existing Content Manager content to LWWCM using federated content components, connect tags, and JSP components. Having covered the basics of how to integrate, we address some of the issues you need to consider when planning the entire integration system architecture. This includes covering different scenarios of having LWWCM and Content Manager in one machine, having multiple LWWCMs using syndication, and having multiple Resource Managers in the setup. The part concludes with a discussion of portal integration.

In Part 3, we cover advanced topics including search, security, integrating VideoCharger™ videos, data migration, and versioning.

This IBM Redbook is intended for IT architects and specialists working on LWWCM and/or Content Manager who will be in charge of an LWWCM and Content Manager integration project.

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 Manager 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 of Sciences 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.

Stephan Bolten is an Advisory IT Specialist of the IBM Software Services department in Zurich, Switzerland. He has more than five years of experience in IT, particularly in the development of IBM Lotus Domino® solutions and IBM WebSphere® Portal Server and IBM Lotus Workplace related applications. His further areas of expertise include Web Content Management solutions with IBM Lotus Workplace Web Content Management for Domino and Java™, enterprise integration and enterprise search approaches. He has worked extensively on projects involving those topics since he joined IBM in 2000.

Ziad Hakim is a consultant with ebf-EDV Beratung Foellmer GmbH in Cologne, Germany. He is an IBM Certified Professional for Lotus and WebSphere. He has over six years of experience in solution design and architecture, as well as in system consulting and integration. He has extensive Lotus and WebSphere installation, deployment and application developer skills and has been working with Lotus Workplace Web Content Management since its first release.

Maria Elena de Leon is a CLP (Certified Lotus Professional). She holds a degree in Computer Science from the Facultad de Ingeniería, Universidad de la Republica, Montevideo, Uruguay. She has been working at ISA Ltda, an IBM Business Partner, since 1999. Her areas of expertise include Apatrix, Lotus Workplace Web Content Management, Domino applications, and Java applications for WML technology. She has more than five years of experience in the development of IBM Lotus Domino solutions and has worked as a consultant on many different kinds of projects, particularly projects related to quality assurance that follow ISO standards. She has also developed customized solutions for international companies in foreign countries like Costa Rica, where one of the main goals was the integration of Lotus Domino Solutions and Siebel.

Walter Mayer is a technical consultant for IBM Content Management Technical Enablement Channels and a member of the IBM DB2 Information Management EMEA team in Germany. He holds a degree in Physics. He has several years of experience in the software development and customer services area as a technical consultant for Enterprise Information Portal and Text Mining. Between 2001 and 2003 he worked as a DB2 and Pervasive Computing (PvC) consultant and DB2 Everyplace® project leader in EMEA. He has published several papers on mobile computing and has taught DB2 Everyplace, PvC and, since 2004, IBM CM Certification workshops throughout EMEA. He has worked on numerous IBM BP projects since he joined the IBM BP EMEA Enablement team in 2000.

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

Maresh Garg
David Sanders
Jerald Schoudt
Arya Adeli
Basuki Soetarman
IBM Silicon Valley Lab, US

Jason Hatch
Rob Enright
IBM Software Services for Lotus, UK

Christoph Rupp
IBM Software Services for Lotus, Switzerland

Stefan Momma
Juergen Metter
Holger Koenig
IBM Boeblingen Lab, Germany

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



Part 1

Introduction

In this part, we provide brief overviews of LWWCM and Content Manager. We discuss why we might want to integrate both products and when to do so. We also introduce a case study, for an insurance company, which we will be using throughout this redbook to show sample integration steps, procedures, and coding.



Lotus Workplace Web Content Management overview

This chapter describes LWWCM's features and benefits and the importance of quick and easy Web site maintenance.

The topics addressed in this chapter are as follows:

- ▶ Introduction to LWWCM
- ▶ Architecture of LWWCM
- ▶ How to create and update Web sites:
 - Information architecture
 - Web page design
 - Collaboration teams for distribution and management of contents
 - Versioning
 - Syndication
- ▶ Retrieval of data from different sources

1.1 Introduction

IBM Lotus Workplace Web Content Management (LWWCM) provides an end-to-end solution that allows users with non-technical skills to easily publish their documents in a Web site. Content creation is simplified and streamlined by allowing users to use reusable components and pre-built templates.

LWWCM separates the design and presentation from content creation. This allows the creation of content once, and the display of the same content with a different look and feel. You can easily create and maintain your Web site without worrying about the impact of the contents on the Web site's look and feel. This also guarantees a consistent presentation because the contents remain unchanged even when the Web site design changes.

Using LWWCM, the development and delivery of information is accelerated; consequently, this drives down the cost of creating and managing Web site contents, which usually exist in different forms and formats within companies. With LWWCM, the information can be freely distributed and instantly updated across all existing applications, including Internet, intranet, and extranet Web sites.

In addition, LWWCM provides the following benefits:

- ▶ Easy-to-use features, allowing users to deploy a Web site faster
- ▶ Scheduled publishing and expiring of Web pages
- ▶ Scalability from mid-sized to large enterprises
- ▶ Collaborative functionalities, including workflow, personalization, and caching services
- ▶ Security settings to control user access to different areas of the Web site
- ▶ Syndication, allowing the back-up and restoring of the published business information
- ▶ Versioning of content, which could be helpful through the document review and approval process
- ▶ Cost reduction of managing sites, while ensuring that information on the sites always remains up-to-date

Figure 1-1 on page 5 shows how content authors and Web page designers can work separately to create the Web page(s) that appear on a Web site.

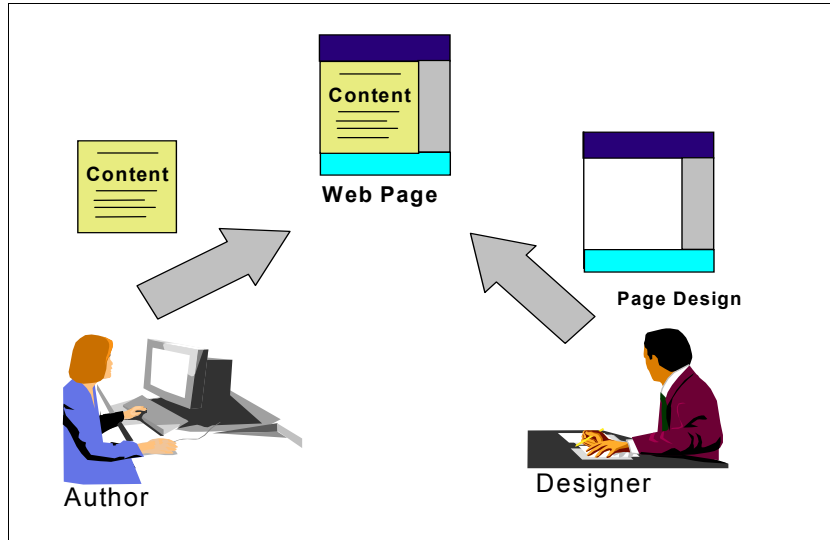


Figure 1-1 Authors and Web page designers working separately

1.2 LWWCM architecture

LWWCM includes team collaboration, messaging, document management, and a variety of other business processes and applications. Within the Workplace environment, LWWCM provides collaborative Web content design, approval, caching, and staging.

Within a Web page, the information can come from multiple repositories. Figure 1-2 on page 6 shows an example of a Web page where information comes from three different sources:

- ▶ The *A* data is populated from the LWWCM application client and is stored in the LWWCM repository that could be IBM DB2 Content Manager CM, IBM DB2, or XML.
- ▶ The *B* data is populated from an external Content Manager application.
- ▶ The *C* data is populated from an external DB2 database.

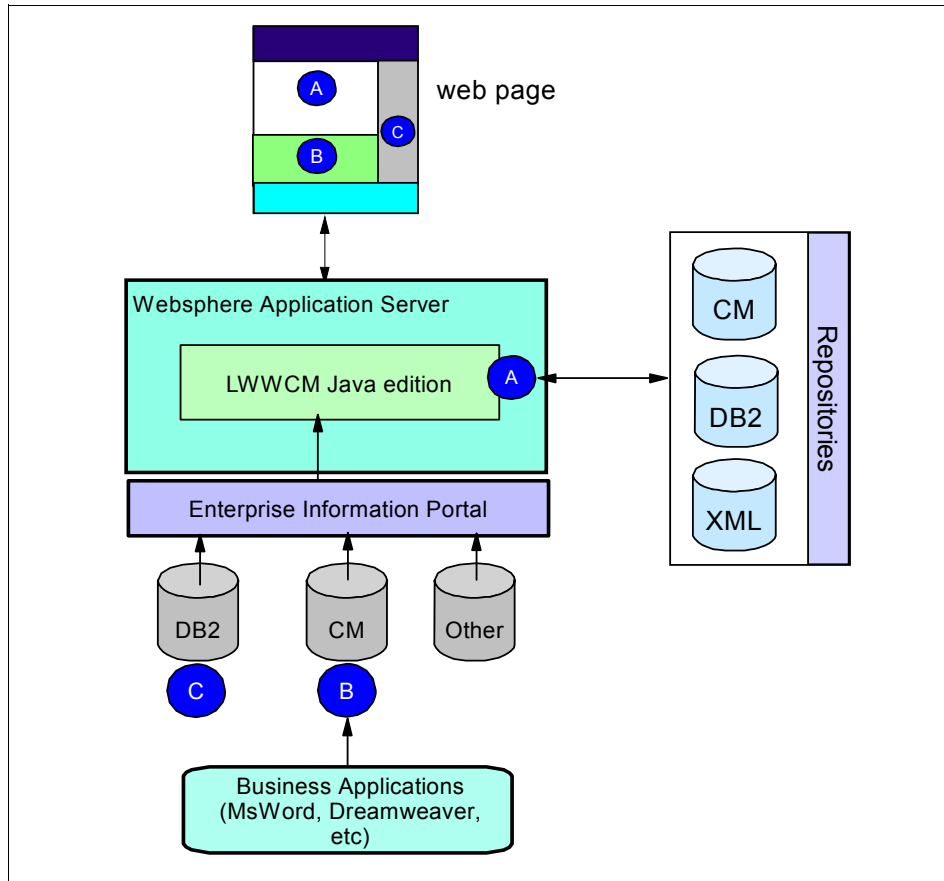


Figure 1-2 LWWCM architecture

LWWCM is available for Lotus Domino and the Java Edition for WebSphere Application Server (including Portal server).

The *Domino edition* consists of a group of Notes databases where the Web page content of the Web site and the Web page design are stored as Domino documents in a publishing database and are rendered when a page is requested.

The *Java edition* consists of a Web application that is installed in a Web application server. The Java edition can use one or more of the following external repositories: IBM DB2 Content Manager (Content Manager), IBM DB2, and Oracle.

Table 1-1 on page 7 compares the differences in features between the Domino and Java editions.

Table 1-1 Comparison between LWWCM Domino edition and Java edition

Feature	Domino	Java 1.1	Java 2.0
Version control	Yes	No	Yes
Batch edition	Yes	No	No
Export site to HTML	No	Yes	Yes
Multi-lingual content	Yes	Yes	Yes
Multi-lingual translation workflows	Yes	No	Yes
Complex templates	No	Yes	Yes
Multiple content clients	Yes	No	No
Content import	Yes	No	Via API

Since the main topic of this redbook is how to deal with the integration of LWWCM with Content Manager, for the remainder of this redbook, we cover LWWCM, the Java edition, V2.0, with Content Manager as the repository.

1.2.1 LWWCM user interface

LWWCM user interface (sometimes also called the LWWCM administration user interface) shows a navigator on the left, and is divided into the following modules: Content Management, Design and Development, Site Management, Security and Workflow, Version Management, and Syndication and Deployment.

The *Content Management* module stores the contents within its Content Library.

The *Design and Development* module stores the design elements and components such as page designs, images, files, content templates and search rules.

The *Site Management* is composed of Site Framework and the Category Management. The *Site Framework* comprises the site structure definition, that is, the Web site map. The *Category Management* comprises the category taxonomy, which is the classification of content in categories.

The *Security and Workflow* module stores the users and groups as well as the workflows, stages and actions.

The *Version Management* module contains the Version Library, which is used to store multiple copies of an object as versions.

The *Syndication and Deployment* module contains the syndicator and subscriber objects to achieve syndication among different servers.

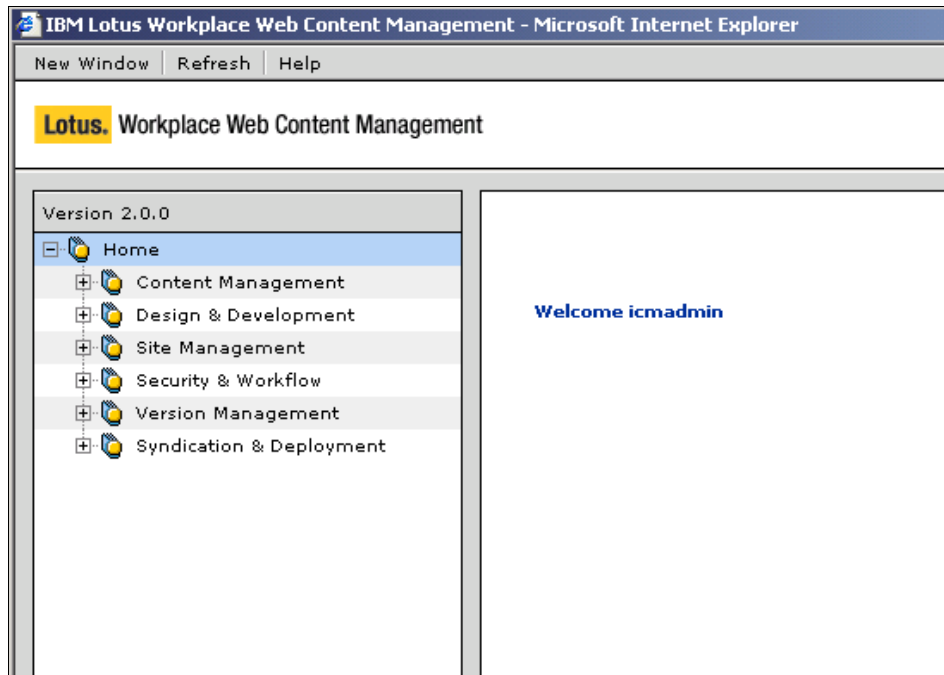


Figure 1-3 LWWCM user interface with left-side navigator

The *Design and Development* module contains components that are mainly used during the design phase of the Web site presentation. These components are configurable and reusable objects such as content menus (based on a search by attribute), navigators, menus, and images. Some of these components are mainly used during the design phase of each Web page, such as *page design* or *content templates*. Other components are directed to the common users during the content creation phase.

For example, a component page design contains the skeleton for a Web page and is included within a *site area*. When a user creates content, the user only has to choose the specific content template and site area. The templates are used to store certain types of default information that are common to all the contents created with the same template. For this reason, the content templates, along with the page's design, help to ensure that the published content is uniform and has the same look and feel throughout.

1.3 Creating and updating Web sites

The components that we must consider in order to create a Web page are the *site area*, *workflow*, and *page design*. As the first step, the Web site developer must create site areas. After that, the Web site developer creates page designs, content templates (which will be included within the proper site area) and workflows. Once they are ready to be used, content creators select the proper content templates, choose an appropriate site area, and create their Web pages.

1.3.1 Information Architecture (IA)

The first thing we need to keep in mind when building a Web site is how the information in our Web site will be organized. This is known as the *Information Architecture* (IA). The IA determines how information on a Web site should be organized and linked in such a way that people find it easy to use and to understand.

In LWWCM, this IA structure is defined in the *Site Framework* and represents the Web site hierarchy. Each entry (site area) in the Site Framework dynamically creates a link to the page representing the proper entry point of a Web site. From the Site Framework, the navigators can be created to be included in the target Web site.

1.3.2 Web page design

A *Web page* is mostly made up of components and content. The components are referenced by the underlying *page design* that builds the basic layout and the default properties (for example: background color, images, and frames) of the Web page. Figure 1-4 shows the page design and components that make up a Web page.

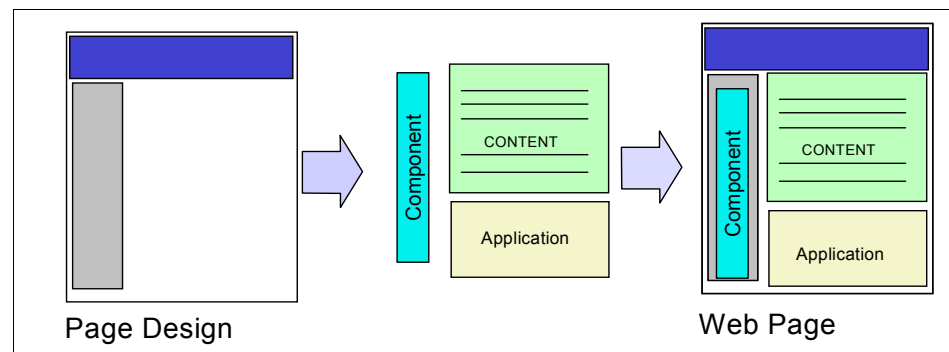


Figure 1-4 Page design and components for a Web page

LWWCM provides its own content creation tool. You can either enter your content as rich text or as HTML directly. There are no out-of-the-box import filters for document types like MS Word, Dreamweaver, etc. You can still create your content with these tools but you would have to save it as HTML and copy/paste it into the editor LWWCM provides.

The contents created with all these different tools are presented on a Web site with little or no modification to the source documents. In fact, LWWCM can present multiple content types on the same Web page.

A *Content Template* contains the data entries for the content components, such as title, summary, and central content body. The only content components that authors can create are those which were defined in the Content Template.

A *Page Design* is composed of HTML code that layout the structure of a Web page and provide some of the general page properties. A Page Design also uses LWWCM tags to reference other specific components.

Components may be divided into *design components* (for example: menus, page design, and navigators) and *content components* (for example: content main body text, content summary, and title).

Design components reside under the *Component Library* of the LWWCM user interface tree view, which is part of the Design and Development module, while the content components reside under the the *Content Library* of the LWWCM user interface tree view, within the Content Management module.

1.3.3 Collaboration for content management and distribution

LWWCM provides an *authorization flow (workflow)* for each content to be published. Applying this collaborative approval process implies an interactive integration between people and contents. The author is responsible for each content; there is no third party participant who might make mistakes in the edition or publication process.

These workflows enable the collaborative process of creating, maintaining and deploying Web content over Web sites to ensure accuracy, authenticity, timeliness and effectiveness of content. It also allows users to create and maintain consistent and interactive content representing large amount of rapidly changing information.

This solution facilitates large teams to work together, jointly where necessary, but independently enough so that they can manage changes to content in their respective work groups until they are ready to share them with the extended contribution team.

A *workflow* can be composed of one or more workflow stages. The number of stages are defined by an administrator according to each company's needs. All workflows must have at least one workflow stage. Every content must go through all the selected workflow stages before it is published on a Web page.

As a content moves through the workflow stages, its status changes from "Draft" to "Live" (viewable on a live Web site) and eventually to "Expired" (when it reaches the expiration date).

A workflow stage also determines the access level of users and groups for Web pages in that stage. Figure 1-5 shows some samples of the workflow stages.

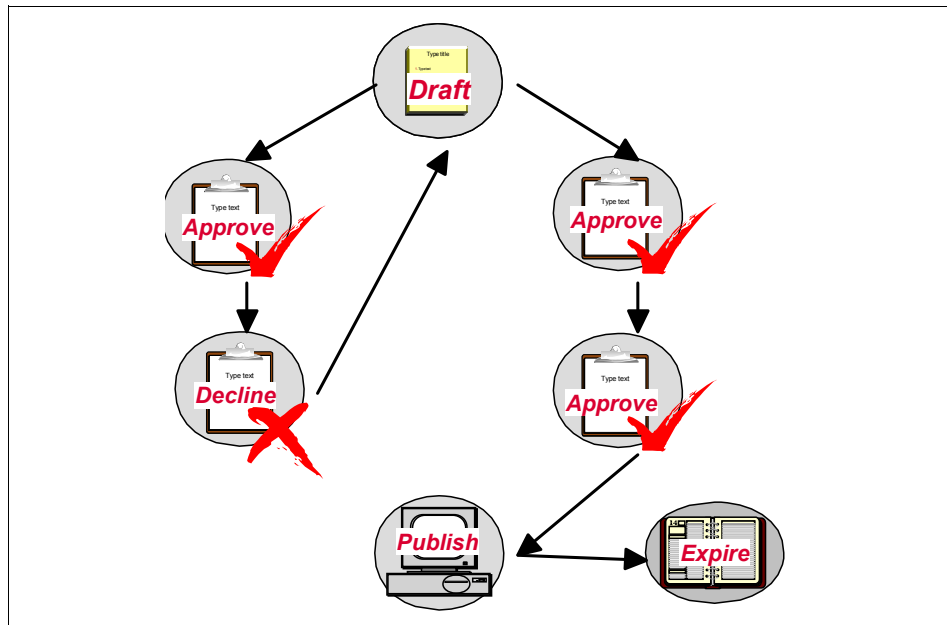


Figure 1-5 Workflow stages sample

When an activity within a workflow is completed, the next activity appears on the task list of the people responsible for that stage in the workflow. This process continues until all the stages in the workflow are completed. At the end of a workflow, the resulting file or document is promoted and published. The approved content then becomes published on the Web site and can be viewed by readers.

1.3.4 Versioning

When you create or update content, you may want to save versions of each item. LWWCM provides the versioning capability for this purpose. If required, you can

return to a previous version of any piece of content. These units are easy to store and to restore.

This topic is described in detail in Chapter 14, “Versioning” on page 293.

1.3.5 Syndication

The LWWCM *syndication* feature is used to replicate data from one server to another.

Syndication is recommended if you have any of the following requirements:

- ▶ Maintain two separate servers for development and production stages, allowing for greater flexibility in development and possibly more efficiency in production without affecting each other.
- ▶ Distribute content across different servers.
- ▶ Migrate content from one repository to another.

1.4 Retrieving data from different sources

LWWCM has the ability to retrieve data from different sources. These sources can be external databases, Content Manager system, file systems, as well as other Web sites. LWWCM can format the retrieved data and display it as part of its own content.

There are different ways to retrieve data from other sources: one can be made in the form of *URL request*; another way is using *Connect Tags* that can be embedded in Page Designs or Component designs. Data Providers are used to retrieve data for display only and this data cannot be modified. The content is *dynamic*, because when the source document changes, LWWCM dynamically updates the content displayed on this page without any intervention.

A Web page can be built from content that is taken from different sources and shown in the same Web page. Note that LWWCM does not take care of the formatting of the source files.

To improve the content's look and feel, LWWCM also allow us to add content and change the presentation of existent content contained in multiple file formats.

Data from Content Manager can be retrieved using either LWWCM *Federated Content Components* or by URL. The Federated Content Components are used to retrieve and store files from federated data sources.

Apart from the federated components, there are *JSP Components*, which are used to store paths to JSP. When JSP Components are included within a page design or other component design, and a Web page using this design is rendered, a request to JSP is generated and processed.

For more information about the retrieving data from different sources, refer to Chapter 7, “Publishing Content Manager content” on page 115.

Figure 1-6 shows an LWWCM architecture sample that breaks down to four area:

- ▶ Creation: Content creation can be implemented using either existing business application tools or with the LWWCM text editor.
- ▶ Storage: Use Content Manager and Tivoli® Storage Manager for content storage.
- ▶ Management: Use LWWCM to manage the content and design of Web pages following approval workflow stages.
- ▶ Delivery: Use LWWCM to publish and deliver the content to Web sites.

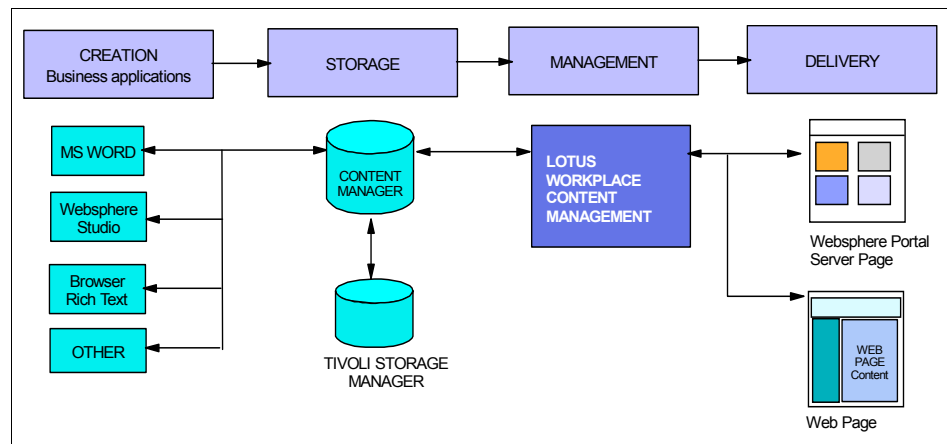


Figure 1-6 LWWCM architecture sample



Content Manager overview

In this chapter, we provide an overview of IBM DB2 Content Manager (Content Manager). We explain why implementing an Enterprise Content Management infrastructure becomes a very important issue for all enterprises in industries and in the public area. We describe the Content Manager product portfolio and Content Manager architecture. We also cover the main functions and features of Content Manager.

Specifically, we cover the following topics in this chapter:

- ▶ Why implement an Enterprise Content Management infrastructure?
- ▶ Content Manager product portfolio components
- ▶ Content Manager architecture overview

2.1 Why implement an Enterprise Content Management infrastructure?

Every day, we search, retrieve and store information which is important to us in our workplace infrastructure. Important information could be a document such as an electronic file or set of files. It could be a large collection of images, videos, audio, computer output, XML and HTML files or electronic records where each record contains a set of electronic and/or physical objects. Mostly, a company infrastructure is designed, collected, built and accumulated over time.

Applications and services are bought or written on top of the infrastructure and together they support the Enterprise Content Management (ECM) needs. Over time, many systems have emerged to manage content, to store and access daily data chunks of structured or unstructured information to get business done. The storage of content is rarely designed for efficiency. In public and government areas, the demand for methods of governance and administration leads to an increasing demand on infrastructure. There is a consistent increase in numbers of guidelines, specifications and standards. Some of the existing record management standards in the public area are:

- ▶ US 5015.2, Records Management toolkit definition for the US government
- ▶ DOMEA, Document Management and Archiving Requirements for German governmental organizations
- ▶ VERS, Victorian Electronic Record Standard
- ▶ UK PRO2, Archival Records Management definition for the UK government

The need to implement robust content management is steadily increasing. The trend is to move away from a string of disparate projects and environments to an infrastructure approach, integrating content with the line of business, Web Content Management, Digital Asset Management, distance learning, mobile Sales Force Automation (SFA), Enterprise Resource Planning (ERP) solutions or other applications to accelerate the benefits across the enterprise. From the financial point of view, as content grows exponentially, it becomes important to reduce the storage costs and the maintenance costs of overlapping, and to deliver content in a controlled manner to the end users. To address this challenge, IBM offers the IBM DB2 Content Manager product portfolio as described below:

- ▶ IBM DB2 Content Manager (Content Manager)
 - Optionally using VideoCharger as one type of Resource Manager
 - Optionally using DB2 UDB Net Search Extender (NSE)
- ▶ IBM DB2 Content Manager Information Integrator for Content
- ▶ IBM Lotus Workplace Web Content Management (LWWCM)
- ▶ IBM DB2 Document Manager

- ▶ IBM DB2 Records Manager
- ▶ IBM DB2 Content Manager OnDemand
- ▶ IBM Tivoli Storage Manager
- ▶ IBM DB2 CommonStore
- ▶ IBM Electronic Media Management System (EMMS)

Figure 2-1 provides an overview of the Content Manager product portfolio.

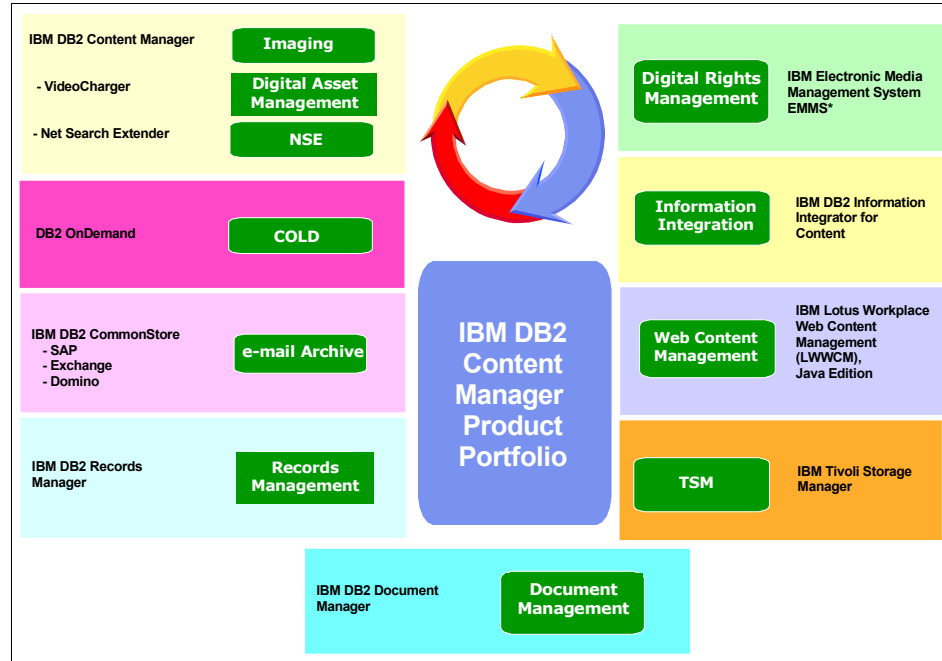


Figure 2-1 Content Manager product portfolio

2.2 Content Manager product portfolio components

In this section, we describe, at a high level, the different components that make up the Content Manager product portfolio as shown in Figure 2-1.

2.2.1 IBM Lotus Workplace Web Content Management (LWWCM)

IBM Lotus Workplace Web Content Management (LWWCM) enables the easy creation of Web sites and Web content. Refer to Chapter 1, “Lotus Workplace Web Content Management overview” on page 3 for an overview of LWWCM.

2.2.2 IBM DB2 Document Manager

IBM DB2 Document Manager (Document Manager) manages the various states and transitions that occur throughout a document's life cycle, from creation to final disposition. Document Manager supports collaborative document creation and management for dispersed user groups, such as creators, reviewers, approvers and contributors. Content Manager can be used as a repository for Document Manager.

The capabilities of Document Manager include the following:

- ▶ Compound document support
- ▶ Bulk document loading
- ▶ Rules-based document life cycle management
- ▶ Desktop application integration
- ▶ Support for engineering file formats
- ▶ Rendition services
- ▶ Remote printing/plotting
- ▶ Process management
- ▶ Automated notification (via e-mail or instant message)

Since Document Manager manages the complete life cycle of business documents, it becomes part of government (e-government) solutions. Document Manager helps government departments reduce paper-based processes and redundant data entries.

2.2.3 IBM DB2 Records Manager

IBM DB2 Record Manager (Record Manager) is used to supply record management solutions through file planning and record life cycle support.

A generic record structure can be considered as a file containing the following:

- ▶ Label
- ▶ Primary electronic and/or physical objects
- ▶ Metadata with pointers to the objects
- ▶ Status and processing information such as status notes and/or annotations

A generic record can be used, for example, to implement a virtual city hall identity card service.

2.2.4 IBM DB2 Content Manager Information Integrator for Content

IBM DB2 Content Manager Information Integrator for Content (Information Integrator for Content) is used to access real time content from distributed sources and deliver that information through a federated interface for applications

or through client solutions. The connection to different back ends is implemented through connectors. The Information Integrator for Content framework has a consistent programming interface across all supported connectors for search and access. The connectors include the federated connector (FED connector) that searches against all federated content and the Content Manager connector (ICM connector) that searches against Content Manager content. With Information Integrator for Content System Administration Client, you can define federated search templates which can be used by the Content Manager application and LWWCM application along with the federated connector to access federated content, including Content Manager data. Refer to Chapter 7, “Publishing Content Manager content” on page 115 for more details on how LWWCM publishes Content Manager content with the connectors and the search templates.

Figure 2-2 shows the architecture for Information Integrator for Content.

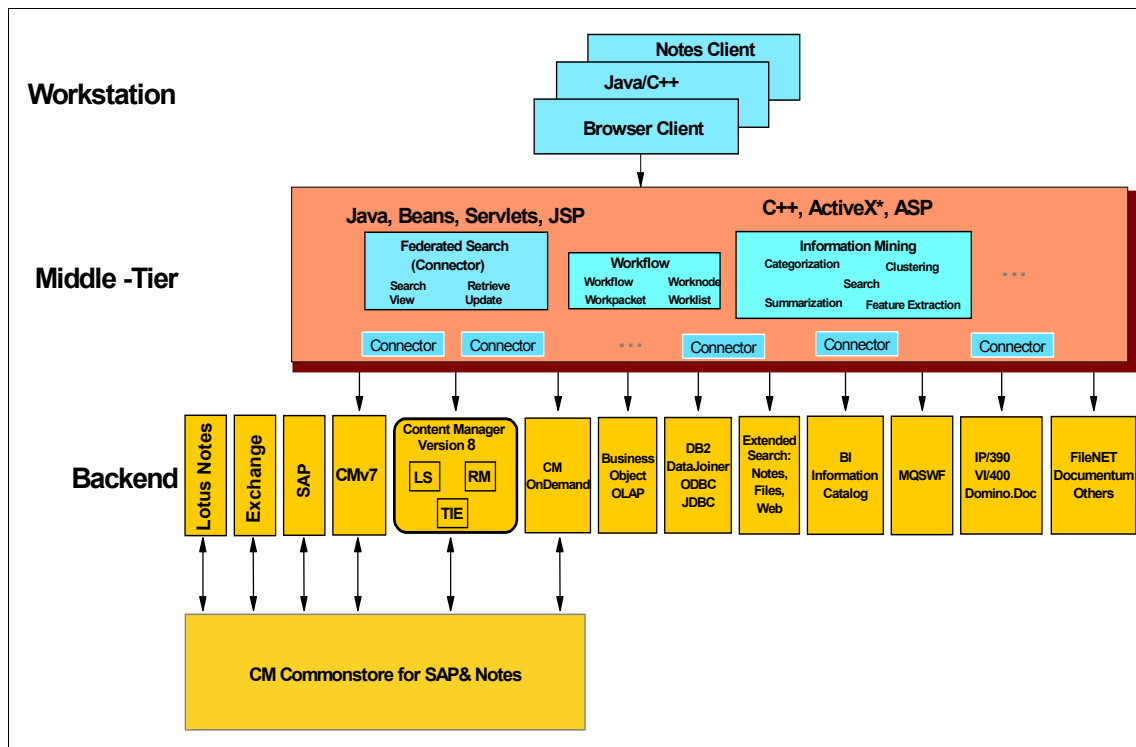


Figure 2-2 Information Integrator for Content architecture

2.2.5 IBM DB2 Content Manager OnDemand

IBM DB2 Content Manager OnDemand (OnDemand) is used to archive the output of “computer output to laser disk” (COLD) solutions/systems. OnDemand stores print spool files and billing reports with efficient query and retrieval capabilities. OnDemand includes the capability to store and manage different document formats for object management along with a company reporting system. Content Manager clients can also search and access the documents archived by OnDemand.

2.2.6 IBM VideoCharger (bundled with Content Manager)

IBM VideoCharger (VideoCharger), bundled with Content Manager for Multiplatforms, is considered one type of Resource Manager in a Content Manager system. VideoCharger manages large media files in the Content Manager system. It stores these files as assets and handles the streaming of these files.

The media types supported by VideoCharger include MPEG1, MPEG2, MPEG, AV, QuickTime, WAV, MJPEG, H263, G723, and LBR.

In our scenario, we use Content Manager as a repository for LWWCM and for video files. In addition, we integrate VideoCharger in the solution such that within a LWWCM hosted Web site, streamed videos can be viewed.

2.2.7 IBM Tivoli Storage Manager

IBM Tivoli Storage Manager (TSM) is a client/server program. TSM is used as an automated, centrally-scheduled, policy-managed archive, back-up and space management facility for workstations and file servers. Within Content Manager, if you plan to provide long-term storage on devices other than fixed disks, TSM is required. For example, if you would like to store objects on an optical jukebox or tape drive, TSM should be installed.

2.2.8 IBM DB2 CommonStore

IBM DB2 CommonStore (CommonStore) archives e-mails, documents, and data from different applications (SAP R/3, Microsoft® Exchange, and Lotus Domino) into different back-end servers (such as Content Manager, OnDemand, and Tivoli Storage Manager). Within Content Manager, the archived e-mails, documents, and data can be retrieved or changed from Content Manager clients. Also, they can still be accessed from within the original application using CommonStore.

2.2.9 IBM Electronic Media Management System (EMMS)

IBM Electronic Media Management System (EMMS) offers digital rights management. EMMS provides an industry-independent foundation for delivery of digital assets that creates new business models, enables flexible digital rights management (DRM) and helps to protect the assets through their entire life cycle.

2.2.10 IBM DB2 UDB Net Search Extender (NSE)

IBM DB2 UDB Net Search Extender (NSE) is an IBM DB2 UDB feature. It is optionally used in Content Manager to add text search capabilities on objects and metadata stored in Content Manager. By using NSE, Content Manager clients or applications can perform searches on document content or Content Manager attributes and other metadata. See 10.1.2, “NSE integration within Content Manager” on page 212 for more details.

NSE search functions (supported by the Content Manager Java and C++ APIs) are:

- ▶ Word or phrase search
- ▶ Boolean search, AND, OR, NOT
- ▶ Proximity search, in the same sentence or in the same paragraph
- ▶ Fuzzy search

These are searches on similarly spelled search terms. The degree of fuzziness can be specified. The first three characters must match.

For example, a fuzzy search on “Shakespere” can return anything with “Shakespeare.”

- ▶ Wildcard search

Use this carefully. It may have a performance impact. There is an internal limit of 1000 for search term expansion.

For example, a wildcard search for “House of p%” can result in 0 hits while a wildcard search for “House of pa%” returns hits that are fewer than 1000.

- ▶ Free text search
- ▶ Ranking
- ▶ Thesaurus search

This is based on a user-defined thesaurus built with the included thesaurus tool.

- ▶ Structured document search (such as XML, HTML, GPP)

- Highlighting of search terms in the search result (the NSE Fixpack 4 and Content Manager Fixpack 6 are needed or else this is not enabled)

2.3 Content Manager architecture overview

In this section, we provide an overview of the *IBM DB2 Content Manager* (Content Manager) architecture and a brief introduction to the main components and features of Content Manager.

A Content Manager system contains a Library Server, one or more Resource Managers and client applications (see Figure 2-3 on page 23). Depending on the Enterprise Content Management solution which is implemented, additional products from the Content Manager product portfolio can be integrated as needed; see 2.2, “Content Manager product portfolio components” on page 17.

Library Server (LS) is a database application that manages users, user groups, access control, data modeling, and document routing processes. Library Server manages the storage and retrieval of all objects’ metadata using stored procedures (SP). It is optionally installed with DB2 Net Search Extender to provide text search capabilities on the stored objects. It has its own database.

Resource Manager (RM) is a Web application that runs on IBM WebSphere Application Server. Resource Manager is a repository for physical files (objects) stored in a Content Manager system. It is a storage management system responsible for storing and retrieving the physical file objects. It has its own database, separate from the Library Server database.

Content Manager clients include System Administrator Client, out-of-the-box clients, and custom clients. *System Administrator Client* is used to set up and configure a Content Manager system. Out-of-the-box clients include Client for Windows® and eClient. *Client for Windows* (also called *Windows Client*) is an out-of-the-box thick client, running on Windows, which enables the end users to search, retrieve, and manage Content Manager metadata and objects. *eClient* is an out-of-the-box thin client, running on a Web browser, which enables the end users to search, retrieve, and manage Content Manager metadata and objects. You can also use C++ APIs or Java APIs provided by Information Integrator for Content to write your own custom client applications.

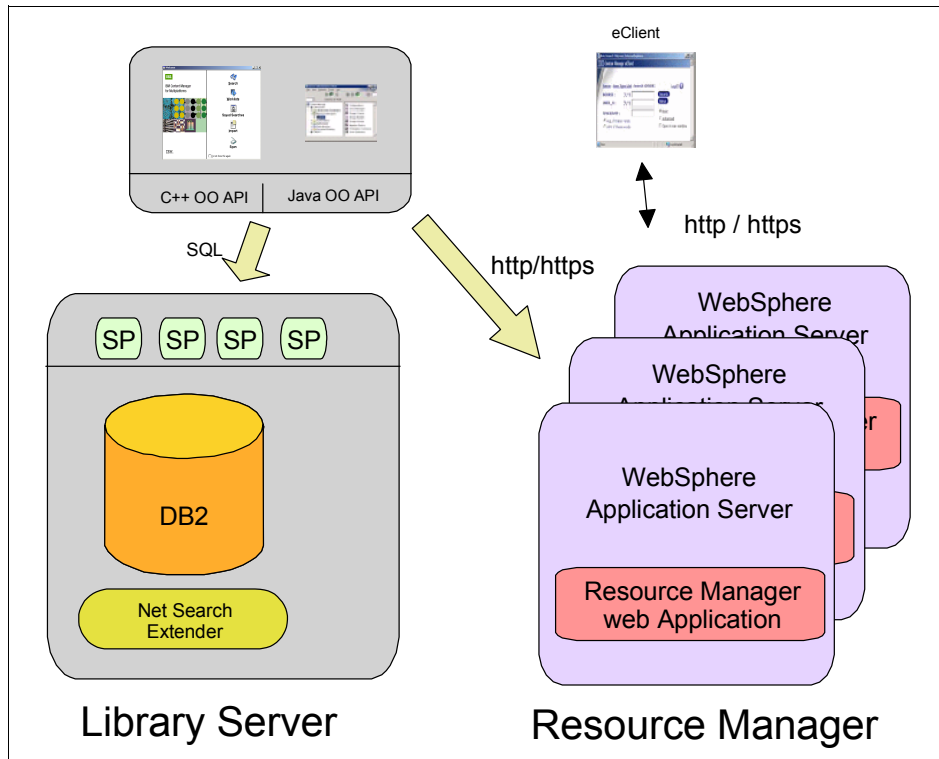


Figure 2-3 Content Manager architecture overview

2.3.1 Library Server

The main component in the Content Manager system is the Library Server (LS). The Library Server is a database application which is implemented with DB2 stored procedures (SP). All the information managed by Library Server is stored in DB2 tables within a Library Server database. Some examples of this information include user definitions, user group definitions, access control information, data modeling definitions, and document routing process definitions. The Library Server application and the Library Server database are always installed on the same machine. The default name of this database is ICMNLSDB. All client access requests to objects stored in a Content Manager System are controlled by the Library Server. Access to the Library Server is through the database query language SQL. On a remote client workstation, the DB2 runtime client needs to be installed to access the Library Server.

Content Manager also includes DB2 Net Search Extender (NSE) to enable a full text search on documents or text metadata in Content Manager.

Figure 2-4 shows the Library Server component.

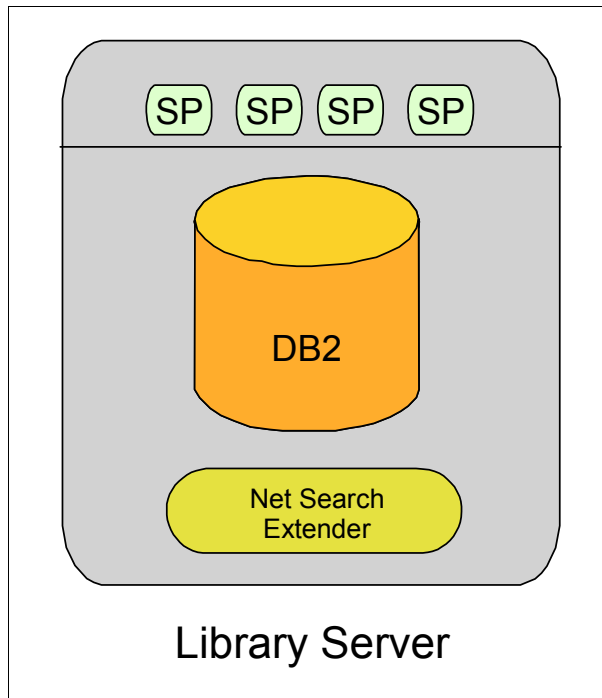


Figure 2-4 Library Server

The System Administration Client is used to specify and configure the Library Server parameters. Use the System Administration Client to define authentication, authorization, data modeling, and document routing.

Authentication

In a Library Server, you define users and user groups that can access a Content Manager system. You can use the Lightweight Directory Access Protocol (LDAP) to import user and group information. Content Manager supports the following LDAP servers:

- ▶ IBM Directory Server
- ▶ Microsoft Active Directory
- ▶ Lotus Domino Directory Notes Address Book

Authorization

When users are defined in a Library Server, they are assigned a privilege set, which is a group of privileges. A *privilege* is a permission to perform an action on an item or object. Some examples of permissions for actions are reading,

modifying, searching, and deleting an item or object. A *privilege set* defines the maximum rights a user has to access a Content Manager system. See Data modeling for information on items.

The *Access Control List* (ACL) defines the maximum rights that can be given to any user against a specific object, where privilege sets represent the maximum privileges granted to a specific user. ACL cannot grant a user more rights than are defined in the user's privilege set, but it can restrict a user's rights. For example, ACL may deny a user the access right to a specific item type object.

Data modeling

The objects stored in a Content Manager system are represented as items. Items are created using templates called *item types*. *Item types* are used to define, import, query and retrieve objects in a Content Manager system.

There are four different *item type classifications* which can be used when creating new item types: item, resource, document part, and document.

Item

This item type is used to store attributes only. There are no objects (such as photos or videos) stored on a Resource Manager. For example, in an insurance company, an item type classified as an item can be used to store the name, address, phone number of all life insurance agents of the company. All item type classifications in Content Manager can have multiple levels of child components within the same classification type as the root component.

Resource

This item type is used for objects with attributes and one and only one corresponding object to store on a Resource Manager. For example, in an insurance company, a resource item type Life_Insurance_Policy can be defined with attributes describing the customer details such as address, phone number, credit card number, insurance policy number and one text document object. You can use this to store the individual contract of a customer.

Document part

This item type stores metadata/attributes and textual, videos, images, photos, notelogs or annotations objects. There are predefined document parts: ICMBase, ICMBaseText, ICMBaseStream, ICMNotelog, ICMAnnotations. For example, annotations can be used to add graphical highlights to an image object. If your application needs this capability, you should define a document part of ICMAnnotations. If predefined parts are not sufficient to design and set up your solution, you can define additional document parts.

Document

This is the item type classification used by LWWCM to store LWWCM objects. It is also *the only item type* supported by the out-of-the-box Content Manager client applications Client for Windows and eClient. A document contains multiple document parts. For example, in an insurance company scenario, we can define a new item type for life insurance contracts named Contracts and add the three predefined document parts, ICMBaseText, ICMStream, and ICMBase. You can import contract objects via Client for Windows, eClient or your own business applications. Contract documents which are text-searchable are stored in the ICMBaseText document part, videos containing additional life insurance policy movies can be stored in the ICMStream document part, and images from the customer can be stored in the ICMBase document part.

Document routing

Document routing is a workflow process that moves documents or folders through predefined steps (work nodes). Users of an integrated solution perform actions to move documents or folders through the process or move them to another process. A defined process needs, at a minimum, a starting work node, one action to be taken by users or an application and an ending work node.

To set up the data model for your solutions, you create new item type(s), set the Item type classification, define metadata, and add the system-managed storage system (SMS) collection. Other things you need to specify include the access control list for the item type, versioning enabling or disabling, logging level, foreign keys, and auto-linking.

To get a better understanding of the Content Manager data modeling, please refer to *Content Manager Implementation and Migration Cookbook*, SG24-7051.

2.3.2 Resource Manager

A Resource Manager is a Web application that runs on WebSphere Application Server. It represents the repository for objects stored in a Content Manager system. You can have one or more Resource Managers but only one Library Server. The Resource Manager Web application uses the DB2 JDBC driver to access both the Library Server database (by default: ICMNLSDB) and the Resource Manager database (by default: RMDB). Figure 2-5 on page 27 shows the Resource Manager architecture.

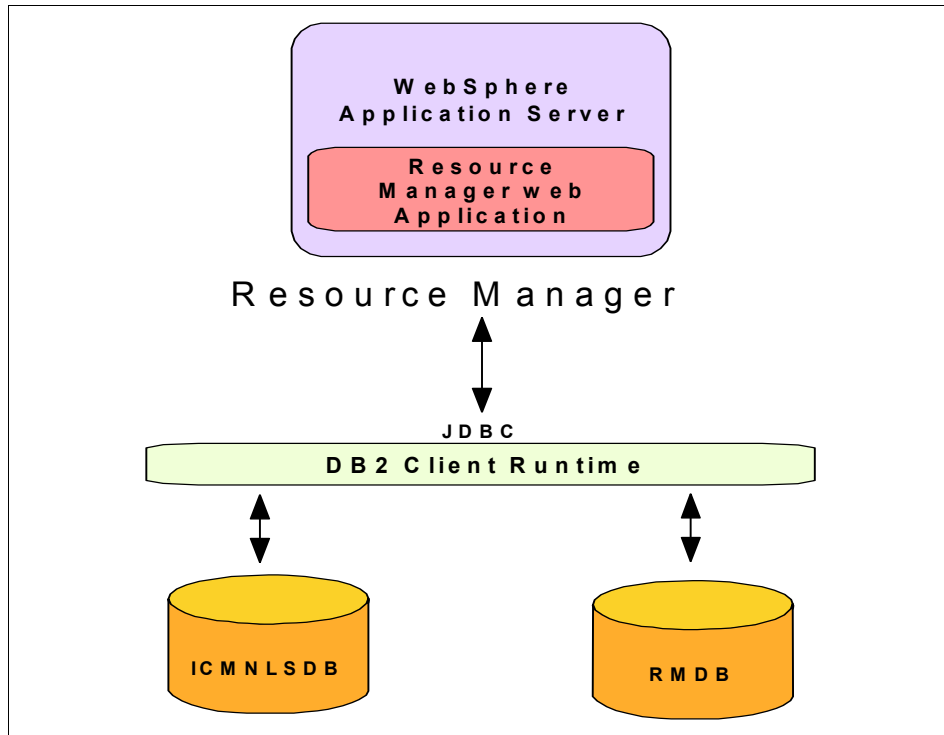


Figure 2-5 Resource Manager

The Resource Manager manages the storage of the physical objects (images, videos, and files) in a Content Manager system. System Administration Client is used to configure the object storage in Content Manager. You have to define where you want to store your object and define if and how you want to move the objects from one storage system to another (*migration policy*).

Storage systems can be:

- ▶ File system
- ▶ VideoCharger
- ▶ Media archive
- ▶ Tivoli Storage Manager (TSM)

For example, you can define the file system and TSM as storage systems and also a migration policy. The migration policy is defined such that the objects stay on the file system for one year, then move (migrate) to TSM and stay there forever. Finally, you define a collection to bind a storage group and a migration policy together. A collection contains where and how long an object is stored and whether it will be transferred. If the object will be transferred, the collection states which storage system it will be transferred (migrated) to. Figure 2-6 on page 28

shows that information stored in Machine 1 will eventually be moved to Machine 2, either onto an optical jukebox or a tape drive. Other archiving options are available as well.

In our integration scenario, we store all LWWCM and Content Manager objects on a local file system forever.

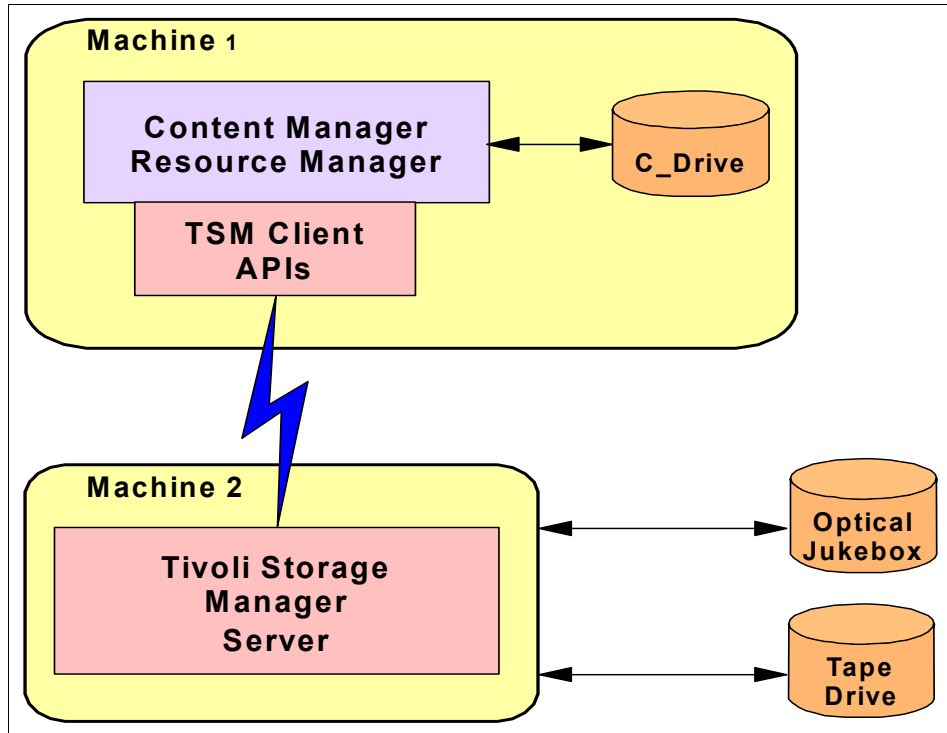


Figure 2-6 Resource Manager manages the storage of the objects

The System Administration Client can be used to configure additional Resource Manager properties such as the start, duration and cycle time of the migrator process. The *migrator* is a thread running in the background that migrates objects from one storage system to another. You can configure to specify when it should be started, when it should stop, and how often it should run. One additional task of the migrator is to reconcile with the Library Server database, and physically delete any obsolete objects from Resource Managers. There are other processes, replicator, stager, and purger, that you can configure within the System Administration Client.

For more information, refer to *Content Manager Implementation and Migration Cookbook*, SG24-7051.

2.3.3 System Administration Client

You have to use the System Administration Client to define and manage your Content Manager System.

From the System Administration Client, you can:

- ▶ Configure your Library Server.
- ▶ Configure your Resource Managers.
- ▶ Define Storage Management Systems (SMS).
- ▶ Define users and user groups.
- ▶ Create privileges, privilege sets and the access control lists.
- ▶ Define your Data Modeling.
- ▶ Define a Document Routing.

Figure 2-7 on page 30 shows the interface of a System Administration Client.

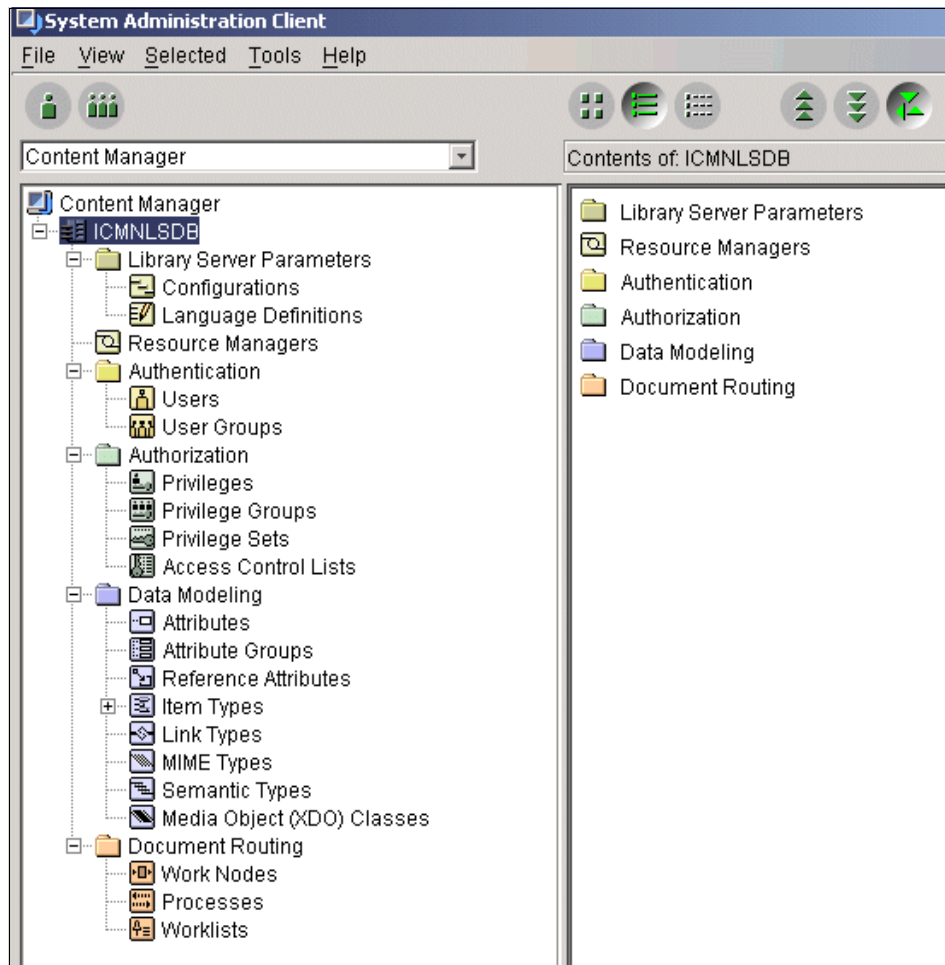


Figure 2-7 System Administration Client interface

2.3.4 Content Manager clients

Content Manager clients can be out-of-the-box clients or custom clients. Out-of-the-box clients include the thick client, Client for Windows (also known as Windows Client), and the think client, eClient.

Client for Windows

Client for Windows is a thick client shipped with Content Manager. It provides an interface for users to work with the data stored in Content Manager. Using the Windows Client, you can import, store, and retrieve objects. This client has to be installed on a Windows system.

eClient

The eClient is a browser-based client that can be run on any system using the Internet Explorer or Netscape Navigator browser. It is a thin client that provides an interface for users to work with the data stored in Content Manager.

With eClient, requests are sent to the eClient Web application, which is installed on the WebSphere Application Server, via http or https. The WebSphere Application Server then makes a connection to the Library Server. With the eClient, you can also access, import, modify, and delete the metadata and/or objects stored in a Content Manager System. The right for a user to manage a particular object depends on the user's privilege set and the object's ACL. Figure 2-8 shows the client access to the Library Server and the Resource Manager.

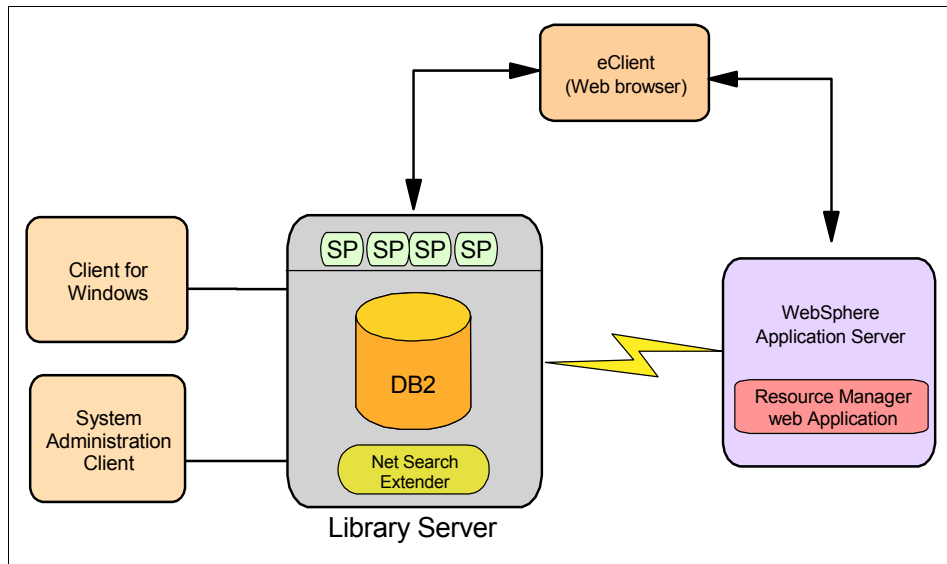


Figure 2-8 Clients access to Library Server and Resource Manager

2.3.5 How they work together

The Library Server, Resource Manager, and the client application work together to manage content in a Content Manager system. To demonstrate how they work together, we provide the following scenario:

1. A client sends a request to the Library Server over a Web application.
2. The Library Server authenticates the user and sees if the user has the rights to have the request fulfilled.

3. If the request is related to metadata only, the Library Server sends the object to the client.
4. If the request is for an object stored in a Resource Manager (such as images, file, audio, video, or document), the Library Server sends a token to the client. The token contains an ID, expiration time, and the location information for the requested object.
5. The client then sends this token to the Resource Manager via HTTP. The Resource Manager validates the token and processes the client request.

Note that the token is valid for 48 hours from the time the client receives it from the Library Server. A portion of the token is the encryption key. Both the Library Server and the Resource Manager should have the matching encryption key. The encryption key can be regenerated (refreshed) from the System Administration Client's Library Server configuration panel. If they do not match, the user will receive an error message.

2.3.6 Working with LWWCM

With IBM Lotus Workplace Web Content Management (LWWCM), you can create and manage your Web site using the LWWCM interface and use Content Manager as the back-end content repository. See Chapter 6, "Using CM as a repository for LWWCM" on page 91 for more details. In addition, you can create and manage data in a Content Manager system and publish the data through LWWCM. See Chapter 7, "Publishing Content Manager content" on page 115 for details.

Figure 2-9 on page 33 shows a diagram of how Content Manager integrates with LWWCM.

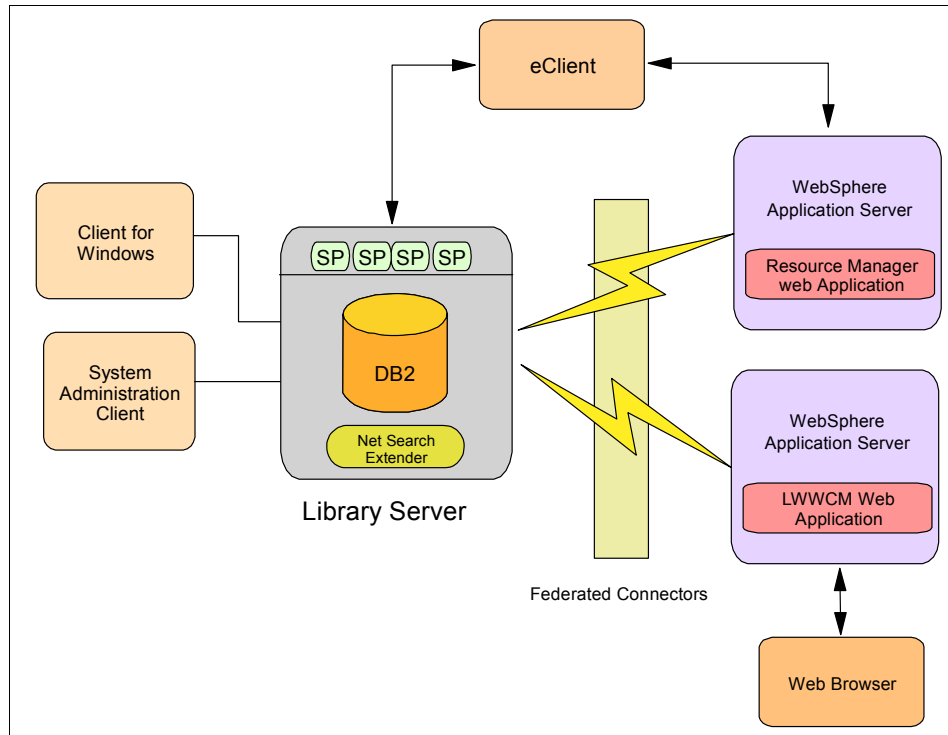


Figure 2-9 LWWCM and Content Manager working together



Why and when to integrate

This chapter discusses why and when to use LWWCM in conjunction with Content Manager.

Specifically, in this chapter, we cover the following topics:

- ▶ Reasons to integrate
- ▶ Benefits of integration
- ▶ Value propositions
- ▶ Advantages of CM compared to other repositories
- ▶ Sample scenario

3.1 Reasons to integrate

There are many business reasons why you would want to integrate IBM Lotus Workplace Web Content Manager (LWWCM) with IBM DB2 Content Manager (Content Manager). Following are some of the reasons when and why you should integrate:

- ▶ Leverage an existing Content Manager infrastructure.
- ▶ Save time and money by re-purposing content instead of re-creating it.
- ▶ Simplify administration of managing content.
- ▶ Manage the expansion of diverse content types, placing different Web assets into the same Web page.
- ▶ Reduce costs: allow reuse of infrastructure components and information.
- ▶ Promote deployment of timely and personalized content.
- ▶ Easily publish content from Content Manager to pervasive devices using LWWCM templates.
- ▶ Take advantage of the fact that any content produced can be easily accessed by non-technical users and approved for publishing on Web sites.
- ▶ Leverage your enterprise content to the greatest possible advantage, streamlining the operations, reducing the costs, strengthening the customer and partner relationships and producing a rapid return on your information.
- ▶ Minimize probability of data leakage.

3.2 Benefits of integration

In general, there are two points of view from which you can look at the integration of LWWCM and Content Manager:

- ▶ Integrating using Content Manager as the repository is discussed in 3.2.1, “Integrating using Content Manager as the repository” on page 37.
- ▶ The integration of external resources is discussed in 3.2.2, “integration of external resources” on page 37.

The bottom line is that the combination of LWWCM and Content Manager enables you to capture, manage and re-purpose all forms of content across diverse applications, business processes, and platforms to deliver integrated, consistent and on demand information to customers, partners, and employees.

3.2.1 Integrating using Content Manager as the repository

All objects created in the LWWCM administration interface can be stored in Content Manager except the unsecured resources which can *also* be stored and retrieved from the file system depending on the *resource.nonsecure* settings in the *aptrix.properties* file. Secured resources (images and files) can only be stored in Content Manager.

Note: For more information on secure and non-secure resources storage, refer to LWWCM's online Information Center (**Installation Guide -> Server Configuration -> Web Content Management Data Storage -> Configuring Resource Storage**).

Having all of your LWWCM objects stored in Content Manager has the following advantages:

- ▶ Security: the data and resources stored in Content Manager cannot be viewed by or edited by unauthorized people, whereas, for example, content stored in a file system is less protected against unauthorized access.
- ▶ Scalability: content Manager manages the stored data. Your LWWCM objects can grow with a Content Manager system.
- ▶ Integrated hierarchical storage management and archiving capability: these are provided when integrating with Content Manager. An example is the Tivoli Storage Manager (TSM).
- ▶ High performance and availability using multiple Resource Managers.
- ▶ Ease of administration compared to other repository types.

3.2.2 integration of external resources

Using the federated content component, the connector tag, or the JSP component, you can easily publish existing content stored in Content Manager into your Web page. Content of all types (such as scanned documents, digitized images and videos) may be stored and managed in Content Manager. You can either display them through a link on the page or render them directly into the final page. Storing this kind of data in Content Manager gives you the full range of benefits Content Manager offers to manage its content.

Note: Refer to Chapter 7, "Publishing Content Manager content" on page 115 for more details.

In addition to the advantages listed in 3.2.1, “Integrating using Content Manager as the repository” on page 37, using Content Manager provides the following advantages:

- ▶ Using Library Server services such as versioning and check-in/check-out.
- ▶ Leveraging a search engine combined parametric and full text search.
- ▶ Federated searches across disparate content stores are treated as one for the end user.
- ▶ Seamless integration with Information Integrator for Content which provides federated access to multiple sources of information (federated search).
- ▶ Ability to link to content creation tools using the Open Document Management API (ODMA) technology, that is, direct integration, for example with word processing applications and high-volume scanning solutions.
- ▶ Retaining benefits of single source content since all data is stored in Content Manager only and therefore is easier to maintain.
- ▶ Advanced security features down to the object level; protecting content through encryption and access control protection.
- ▶ Single Sign-On (SSO) with WebSphere application server.
- ▶ Integrated basic workflow if Information Integrator for Content is used.

Content Manager provides richer content management functionality than the file system and database repository options for LWWCM. By deploying LWWCM on top of Content Manager, you may extend content management functionality beyond pure Web content management throughout your organization.

3.3 Value propositions

In this section, we discuss the value propositions of Content Manager, LWWCM and of the integrated solution, using both products together.

3.3.1 LWWCM

LWWCM puts content management back in the control of the non-technical business users. It decreases IT involvement (reducing Web maintenance, labor, publishing and training costs) and increases sales and profits by deploying timely and personalized content and increasing stickiness of Web sites. LWWCM improves efficiency in getting all content produced in an enterprise out to the Web. It improves Web site control and quality.

3.3.2 Content Manager

Content Manager provides a unified Enterprise Content Management (ECM) framework for the management of the broad range of content, including Web content, documents, graphics, video, audio, computer-generated reports and presentations. It integrates with existing IT investments, enabling an organization to leverage a common infrastructure and achieve operational efficiencies in the management of all forms of digitized content.

3.3.3 Integrated solution

An integrated solution combines the collaborative authoring of Web content management with the ability to integrate content across diverse applications and processes, promoting reuse instead of re-creation of content stored in Content Manager. It gives the business users control over what is published to a portal, Web site or pervasive device. Any content produced in an organization can be personalized and targeted to individuals or Web sites, or used in marketing campaigns for maximum user experiences and effectiveness. When combined with WebSphere Portal, it also provides one customizable place where a person can perform all work activities, including easily accessing all information on demand.

This solution is delivered on a reliable, scalable architecture based on open standards.

3.4 Advantages of CM compared to other repositories

One question that may arise is the following: why not use other repository types such as file systems or repositories which can be accessed via a JDBC interface (such as DB2 or Oracle)? What are the advantages of using Content Manager as a data repository as compared to other repositories? The following list provides the reasons why you should consider using Content Manager (CM) as a repository for LWWCM:

- ▶ Content Manager provides link management for complex relationships among metadata, which is missing in relational databases.
- ▶ Content Manager provides an embedded workflow engine, which is missing in a relational database.
- ▶ Content Manager provides for management of versions, which would need to be coded into a relational database.
- ▶ Content Manager provides granular access controls down to the item level, which would be difficult in a relational database.

- The Content Manager product portfolio incorporates rich functionality not found in relational databases. This includes streaming, annotation capability, digital rights management, document life cycle management, records management, hierarchical storage management, retention policies and compliance with standards.

3.5 Sample scenario

A company has many branch offices geographically dispersed. Each branch has a separate intranet, which causes a great number of administration difficulties. The company employs thousands of employees. Employees sometimes find it difficult to find information from local Web sites and corporate Web sites. Information comes in many forms: text, images, contracts, billing statements, photos, audio, and video. Employees need to communicate with their clients and each other online for better collaboration and faster speed to market. They need an easy way for non-technical users to manage content for the Web while decreasing training and leveraging an integrated user interface for their many business activities.

The LWWCM/Content Manager solution helps to solve these problems because the integrated solution allows better communication, collaboration with others, ability to search, repurposing and personalization of content throughout the company, and ability to publish the information to a Web site. The result is a greater speed to market, increased productivity, and decreased costs.



Case study

In this chapter, we introduce our case study, an insurance company's Web site, which we will use throughout this redbook. For those who are not familiar with LWWCM, we also provide the steps required to set up this case study Web site.

We cover the following topics in this chapter:

- ▶ Introduction
- ▶ Adding the insurance option into the navigator
- ▶ Creating content for the new insurance options
- ▶ Retrieving data from external sources into insurance content
- ▶ Troubleshooting

4.1 Introduction

The purpose of this redbook is to show you how to integrate Content Manager and LWWCM. Throughout the entire redbook, we use this case study as the business scenario whenever we need to provide specific examples. This case study allows us to demonstrate how to apply theoretical concepts to real world situations.

The case study is based on the *IT JumpStart intranet Web site*, which you can get from your IBM representative and later this year from the Partnerworld IBM Internet site, as a sample of learning and instruction of our integration work.

The Web site is designed for an insurance company. This company offers a wide variety of insurance policy choices for individuals, families and businesses, ranging from life insurance to re-insurance.

One of the most important issues for the insurance company is that insurance policies have many variables which are constantly subject to changes and modifications. For example, the expiration date, validity dates, rates, and special offers are frequently changed. The different plans and policies are distributed in a variety of ways. The authors of this information are distributed among various agencies within the company. The company must keep their customers informed of the updated information.

This has become a critical issue because maintaining and providing accurate and timely information is one of the key benefits that the insurance company is providing to its customers and yet this has become increasingly more difficult to address due to constant changes from a variety of sources.

In order to solve this problem, we propose the use of LWWCM to include all of these plans into one Web site. This way, the information can be modified centrally and frequently. It is easy to add new information without worrying about the impact of this new information on the rest of the Web site.

The IT JumpStart sample intranet provides the following:

- ▶ A set of templates and components which can be modified and added to
- ▶ A demonstration platform with a variety of pre-built pages, menus and navigators
- ▶ A training platform

In order to simplify our case study, we add the insurance option into the IT JumpStart's pre-built menu, which contains the following main menu options: About, News and Events, Strategy and Reports, Admin and HR, Sales and Marketing, and IT. See Figure 4-1 on page 43 for the main menu window.

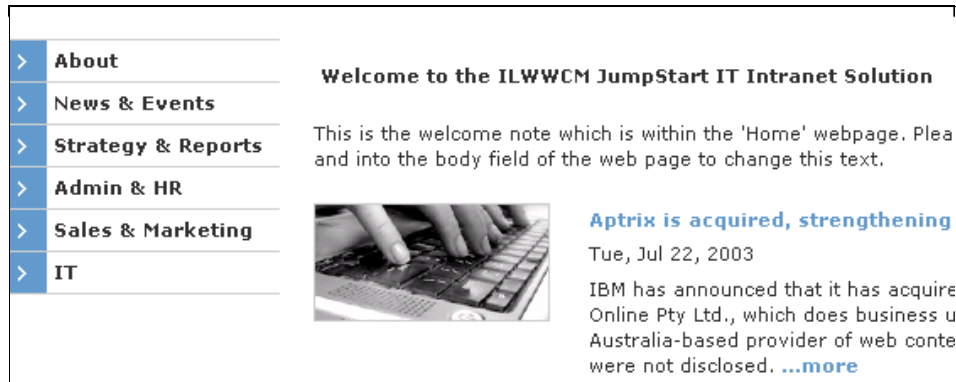


Figure 4-1 IT Jumpstart original main menu

Note: To install the IT JumpStart sample site, obtain the Jumpstart_V1.1.zip file which you can get from your IBM representative and later this year from the Partnerworld IBM Internet site, and follow the instructions given in the index.html file within the zip file.

Once IT JumpStart is installed, we want to customize the Web site to set up the framework for our integration scenarios.

In the following sections, we show you the steps required to customize the Web site for our case study. We assume you have a basic knowledge of LWWCM. If you find it difficult to follow, we recommend that you read Chapter 1, "Lotus Workplace Web Content Management overview" on page 3 and use the following redbook for reference: *Lotus Workplace Web Content Management*, SG24-6309.

4.2 Adding the insurance option into the navigator

Each page of content is assigned to a site area. This in turn defines the page's location in the site framework. A navigator is a snapshot of a section of the site framework. The navigators are built based on the site framework structure.

In this section, we want to add a new insurance option under the existing IT main option in the Web site navigator. To add the new option, we need to add a new site area within the LWWCM site framework.

Creating a new site area

The steps to define a new site area for insurance option are as follows:

1. Start the LWWCM user interface from your browser:
 - a. Type `http://<hostname>:9080/ilwwcm/connect/aptrix`.
 - b. Enter the user ID and password. By default, the user ID is administrator, and the password is password.
2. Click **Site Management** from the interface navigator.
3. Select **Site Framework**.
4. Click **New**. A dialog box opens allowing you to select the site or site areas.
5. From the New Site dialog box, select **Site Area** from the drop-down menu and click **Next**.
6. Select the **Intranet -> Home** site area from the parent list (see Figure 4-2) and then click **OK**. The parent list contains the previously defined sites and site areas. We create a new one at the end of this list.

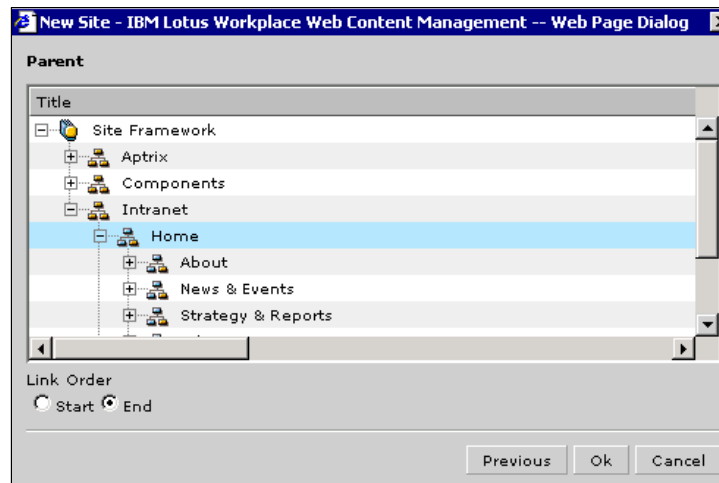


Figure 4-2 New site area

7. In the untitled Site Area Property window, enter Insurance Products in the Name field (see Figure 4-3 on page 45).

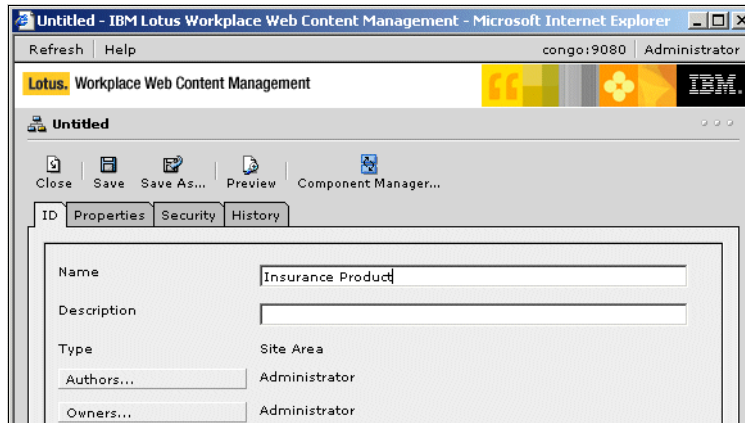


Figure 4-3 Site area properties

8. Click **Save** and then click **Close**.
9. Go to the IT JumpStart to see the changes:
<http://<hostname>:9080/ilwcm/connect/Intranet>

As a result, you should have the new option at the bottom of the original navigator, as shown in Figure 4-4.

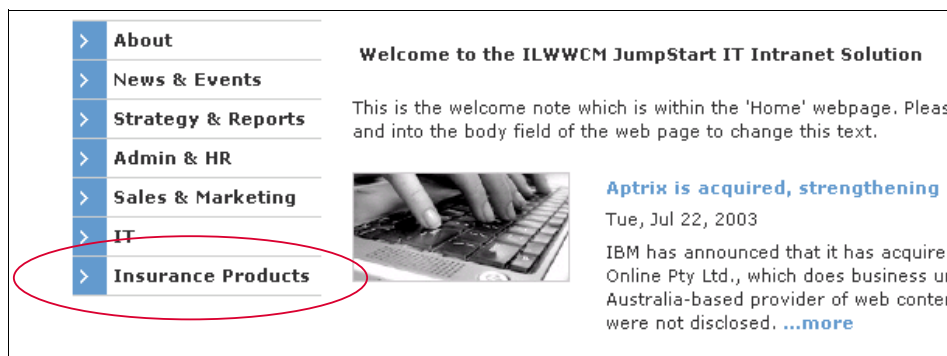


Figure 4-4 Navigator with insurance products option

Note that this menu option is not pointing to any content page. If you click it, no content is displayed. In order to show some content in this menu option, we have to create the content objects from the LWWCM Content Library and then associate them with the site area object's proper content. In the later part of this chapter, we show you how to create content and then associate it to different site areas.

4.2.1 Creating new options under the insurance option

In our case study, the insurance company divides its products into two main groups. One group of insurance policies is directed to individuals and families, and the other group is directed to companies. We need to add two options, under the *Insurance Products* option, one for each group of plans.

To add these two options, repeat the steps described in “Creating a new site area” on page 44 to create all the site areas below the Insurance Products. Make sure that you select the correct parent under which the site areas are located. For example, in this case, the parent of these two new options is *Insurance Products* (see Figure 4-5).

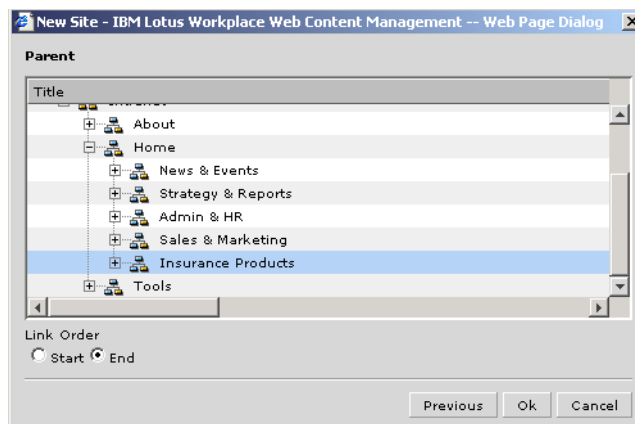


Figure 4-5 Insurance parent site area

We call the first group *Personal and Families* and the second group *Companies*. Create the two new site areas with these names.

Figure 4-6 on page 47 shows the new updates under the Insurance Products menu.

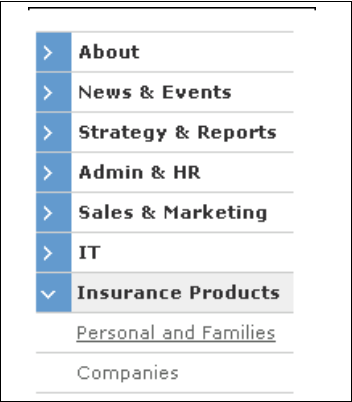


Figure 4-6 Insurance sub-menu

There are different types of plans for each group. These types of plans also must be added into the site framework as different site areas below the appropriate parents. For example, we must create a new site area called *Life Insurance* and place it below the *Personal and Families* site area. Figure 4-7 shows the final structure layout.

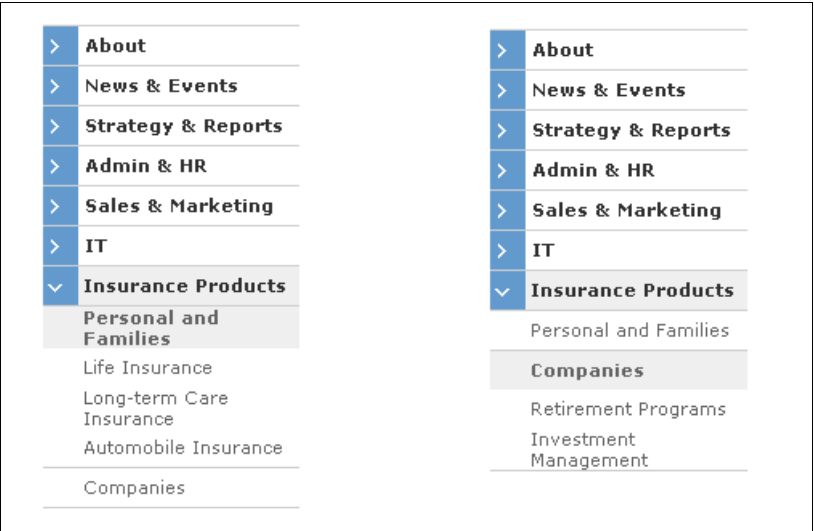


Figure 4-7 Group plans sub-menus

Create the following under the Personal and Families site area:

- ▶ Life Insurance
- ▶ Long-term Care Insurance
- ▶ Automobile Insurance

Under the Companies site area, we create the following:

- ▶ Retirement Programs
- ▶ Investment Management

This is the way in which we create the entire site structure: first the main topic, then the groups, and finally, the different plans inside the groups.

Before testing the new menu options, we have to add content to them.

4.3 Creating content for the new insurance options

We need to create the contents to link them up with the navigator options created in the previous section. In order to maintain the same format throughout the Web site, we first have to create a content template and place in it all of the components which we use in the insurance Web site. This means that if we placed two *image components* in a template, we would be able to include two images in each of the insurance Web site's pages.

4.3.1 Insurance workflow

Before creating a new content template, we need to create a workflow in order to associate the new content template with the specific workflow. This way, we ensure that every page created from the content template uses the same workflow specified within the content template object.

We create a short workflow called *Insurance workflow*.

Creating a new workflow

To create a new workflow, follow these steps:

1. In the LWWCM interface, select **Security & Workflow-> Workflows, Stages & Actions**.
2. Select **Intranet Workflow** (or another existing workflow), and click **Edit**.
3. Click **Save As**.
4. Enter Insurance Workflow as the name (see Figure 4-8) and click **OK**.

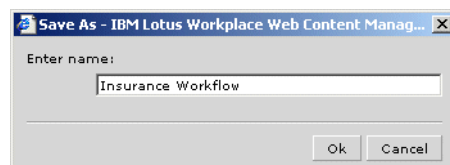


Figure 4-8 Save workflow

5. Select the **Properties** tab.
6. Click **Workflow Stages**.
7. The Workflow Stages pop-up window appears. From the right-hand Selected panel, select the **Intranet Approval** stage and delete it by clicking the <- arrow button (see Figure 4-9).

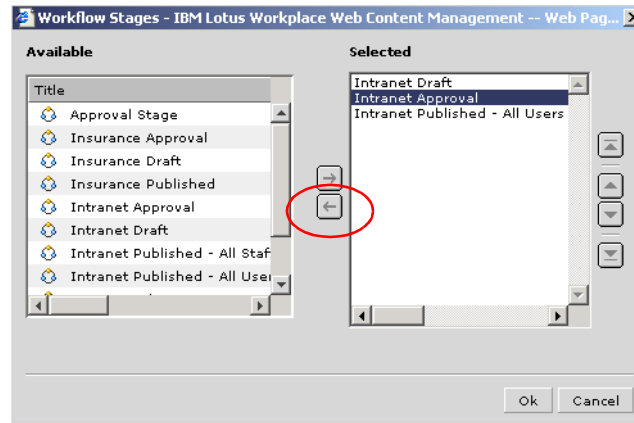


Figure 4-9 Workflow stages update

8. Click **OK**.
9. Click **Save** and then **Close**.

The new insurance workflow contains two stages: *Intranet Draft* and *Intranet Published*. This is to simplify our case study.

When the content is first saved, it will have the Intranet Draft status. After this stage, the content will have the Intranet Published status.

4.3.2 Content templates

Content templates are used to create site content. They contain the fields which the end users (authors) use to provide information to be published into a Web page.

Developers (Web site designer) can set default values for the content in these templates so the end users do not need to enter them. Developers can also include a prompt in a text or rich text field to explain to the end users what type of information is required in a field.

Each field must be matched to a component in a content template. In order to create a new content template, we first have to decide which components (fields)

we need to include within it. In the case study, we decided to use the following components:

- ▶ One *Text Component*, called Title.
- ▶ One *Rich Text Component*, called Body. This is where the main page content text will be entered.
- ▶ Two *Image Components*, called Image and Image2. They are used to upload pictures or images.

All the pages that have been created based on this template will contain the same components specified within this template. Each component will be shown as a field where users can input information.

Creating a new content template

To create a new insurance content template and add the components mentioned earlier, follow these steps:

1. In the LWWCM interface, select **Design & Development -> Content Templates**.
2. Click **New**.
3. Enter Insurance Template in the Name field (see Figure 4-10).

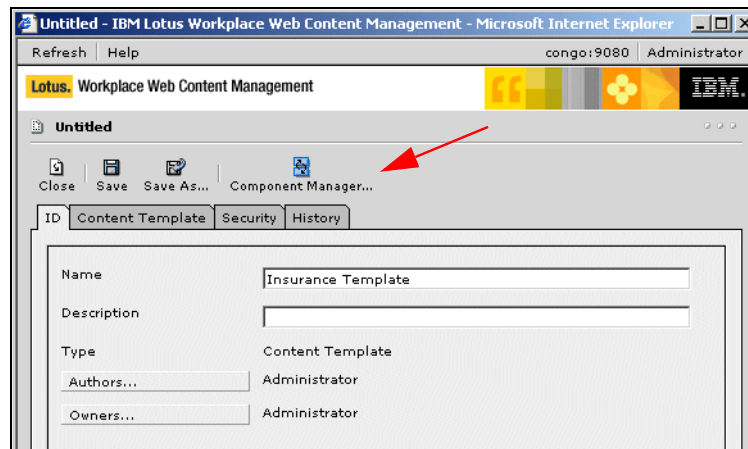


Figure 4-10 New workflow content template

4. Select the **Content Template** tab.
5. Scroll down until you see the Workflow button and click **Workflow**.
6. Select **Insurance Workflow** and click **OK**.

7. At the top of the form, click **Component Manager** (see Figure 4-10 on page 50 where the arrow is pointing).
8. In the Component Manager pop-up window (see Figure 4-11), add the different components we specified earlier into the content template:
 - a. Click **Add**.
 - b. In the Add Component pop-up window (see Figure 4-12), enter **Title** in the Name field and select **Text Component** from the drop-down box for the Type field.
 - c. Click **OK**.
 - d. Repeat the above steps for the two image components (named Image and Image2) and one rich text component (named Body). The result is shown in Figure 4-13 on page 52.

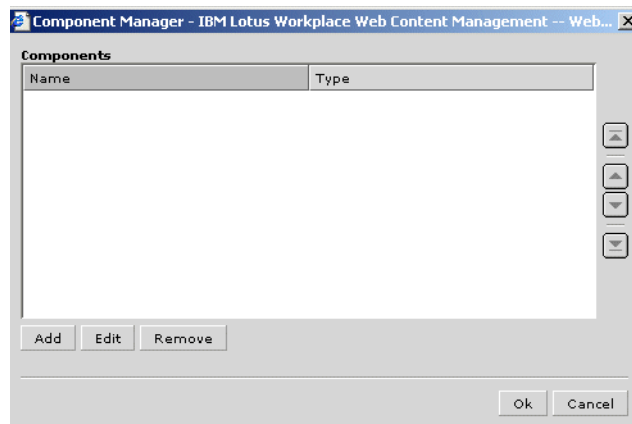


Figure 4-11 Component Manager

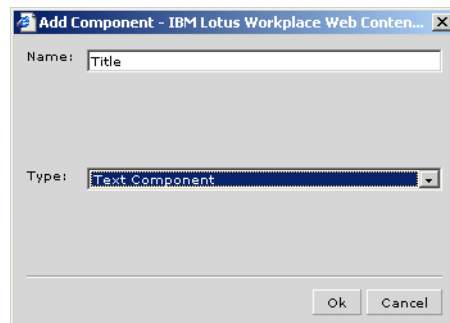


Figure 4-12 Add component in Component Manager

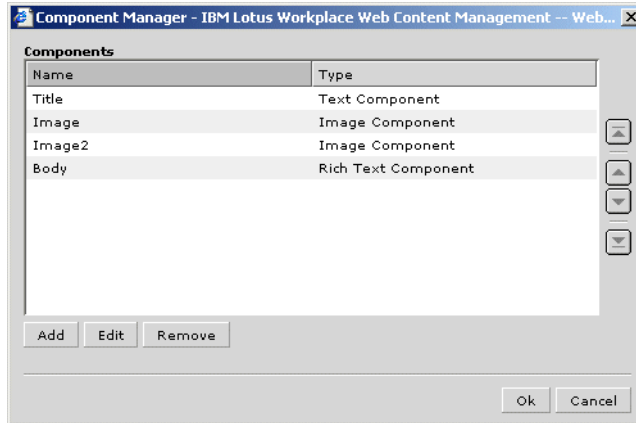


Figure 4-13 Result of after adding all components in insurance template

9. When all the components have been added, click **OK**, then **Save**, and then **Close**.

Figure 4-14 shows part of the resulting form: the first two components, *Title* and *Image*. You can scroll down to see the rest of the components on the form.

The image shows a web form with tabs at the top: "ID", "Content Template" (selected), "Security", and "History". The "Content Template" section is active. It contains a "Title" section with a large text input field. Below it is an "Image" section with several input fields: "Border" (with a value of "0"), "Width", "Height", "Alternate text", "HTML Tag Name", and "File" (with a "Browse..." button next to it).

Figure 4-14 First part of the content template components

Figure 4-15 on page 53 shows the content of each component if you created content using the steps shown in "Creating content" on page 57. Note that the title component, image components, and the body components are identified and circled on the Web page.

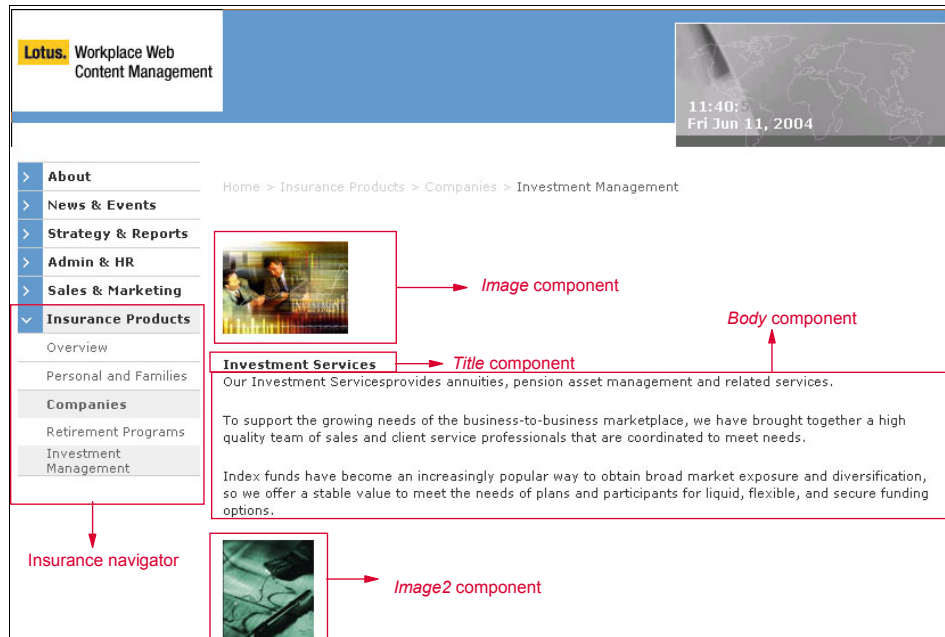


Figure 4-15 Insurance Web page with components added in insurance template

Although the components are included in the content template, the form and framework in which these are shown within a Web page are determined by the page design.

4.3.3 Insurance page design

The content of the page design and components are combined to form a complete page.

Every page design can be applied to many content files. Content is separated from the layout, making maintenance of the site flexible.

In our case study, we take the pre-built page design from the IT JumpStart and we modify it in order to add our new components as mentioned in the previous section. See Example 4-1 on page 54 for the modified Body tag in the page design. The bold HTML tags in the example are what we added to replace the existing Body tag in the original page design, to show the new insurance components within the Web page design.

Note: Anytime we refer to add tags in a page design code, you should replace the existing *Body* tag with our new tags.

The existing Body tag is:

```
<ApatrixCmpnt context="current" key="Body" type="content">
```

In this scenario, our new tags are:

```
<br><ApatrixCmpnt context="current" key="Image" type="content"/><p>
<br><ApatrixCmpnt context="current" key="Title" type="content"/><p>
<br><ApatrixCmpnt context="current" key="Body" type="content"/><p>
<br><ApatrixCmpnt context="current" key="Image2" type="content"/><p>
```

Example 4-1 Sample body tag: Bold tags show insurance components

```
<html>
<head>
<title><ApatrixCmpnt context="current" key="Display Title"
type="content"/></title>
<ApatrixLibCmpnt name="HTML - Stylesheet"/>
<ApatrixLibCmpnt name="HTML - JavaScript Includes"/>
<!-- <ajpetemplate>Detail</ajpetemplate> -->
</head>
<BODY LEFTMARGIN=0 MARGINWIDTH=0 TOPMARGIN=0 MARGINHEIGHT=0
onLoad="showClock()">
<a name="top">
<!-- page header table -->
<table width=100% cellpadding=0 cellspacing=0 border=0 height=142>
<tr><td align=left valign=top>
    <ApatrixLibCmpnt name="HTML - Header"/>
    </td> </tr>
</table>
<!-- page body table -->
<table width=100% cellpadding=0 cellspacing=0 border=0
class="middleBackground" ><tr>
    <!-- nav col -->
    <td align=left valign=top width=10 class="leftPaneBuffer">
        <ApatrixLibCmpnt end="" width='10' height='100'&gt;" name="1pixel"
start="&lt;img src='"/>
        </td>
        <td align=left valign=top width=160 class="leftPane">
            <ApatrixLibCmpnt end="" width='160' height='1'&gt;" name="1pixel"
start="&lt;img src='"/>
            <ApatrixLibCmpnt name="Nav - Left Hand"/><br>
            <ApatrixLibCmpnt end="" width='1' height='50'&gt;" name="1pixel" start="
&lt;img src='"/>
            </td>
            <!-- left margin column-->
```

```

        <td align=left valign=top width=20>
        <AptribLibCmpnt end="" width='20' height='1'&gt;" name="1pixel"
start="&lt;img src='"/>
        </td>
        <!-- content col -->
        <td valign=top align=left width="100%" >
            <table width=100% cellpadding=0 cellspacing=0 border=0 >
                <tr>
                    <td valign="top">
                        <AptribLibCmpnt end="" width='100' height='25'&gt;" name="1pixel"
start="&lt;img src='"/>
                    </td>
                </tr>
            </table>
            <table width=100% cellpadding=0 cellspacing=0 border=0>
                <tr>
                    <td valign="top">
                        <AptribLibCmpnt name="Nav - Breadcrumbs"/>
                    </td>
                </tr>
                <tr>
                    <Td>
<font Class="contrastHeading"><AptribCmpnt context="current" key="Summary"
type="content"/></font>
                    </td>
                </tr>
                <Tr>
                    <td>
<br>
<br><AptribCmpnt context="current" key="Image" type="content"/><p>
<br><AptribCmpnt context="current" key="Title" type="content"/><p>
<br><AptribCmpnt context="current" key="Body" type="content"/><p>
<br><AptribCmpnt context="current" key="Image2" type="content"/><p>

<AptribCmpnt context="current" key="Menu" type="content"/>

```

The tag determines where the referenced component will be shown into the Web page. For example, according to the bold text from Example 4-1 on page 54 above, the *Image* component appears before the *Body* component and the *Image2* component appears after the *Body* component.

Creating new page layout

To create a new page layout, follow these steps:

1. In the LWWCM interface, select **Design & Development -> Page Design**.
2. From the right-hand side, select the **Menu Layout** page design.

3. Click **Edit** to open it in edit mode, modify as described previously, and then click **Save As**.
4. On the pop-up window, enter Insurance Layout.
5. Click **OK**.
6. Click **Save** and then **Close**.

4.3.4 Creating insurance content

A site area defines a content template paired with a page design. When content is created, a content template and a site area must be selected. This selection automatically defines which page design should be used.

Figure 4-16 shows the paired template and page design for our insurance case study. The site area shown is the *Personal and Families* insurance site area, and associates the *Insurance Template* and the *Insurance Layout* with the content template and page design respectively.

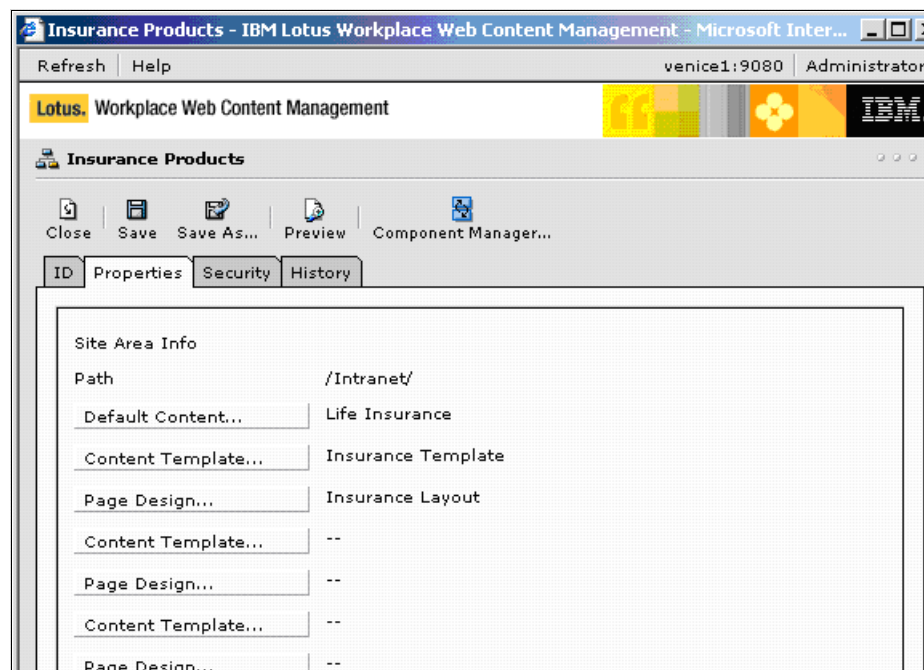


Figure 4-16 Site area properties setting

Content is the only object that end users create. Besides the site area, specifying the proper workflow is also essential for content publishing. It can be the default that is set in the content template, or it can be selected by the end user.

Now that the main LWWCM objects have been created, we can begin adding content to our Web site. We add information to the Name and Description fields as well as to the content components created earlier.

Creating content

To create new content for the insurance company Web page and link it into the proper menu option, follow these steps:

1. In the LWWCM interface, select **Content Management -> Content Library**.
2. Click **New**. Select the **Insurance Template** template and click **Next**.
3. In the available list, expand the list to **Intranet -> Home -> Insurance Product -> Personal and Families**, and select **Life Insurance**. Click the -> arrow to move it to the Selected list.

Tip: Double-clicking the entry will move it to the Selected list automatically.

4. Click **OK**.
5. In the Untitled window, enter Life Insurance in the Name field.
6. Click the **Contents** tab.
7. In the Display Title field, enter Life Insurance. The Display Title appears at the top of the content page.
8. Click **Browse**, located in the Content Image component. Select an image file and click **Open**. The path and file name will be displayed in the file field.
9. Scroll down to the Body field and enter text. Use the rich text editor to format your text. This text will become the content body for the Web page.
10. Repeat Step 8 and add a different image file.
11. Click **Save**.

Referencing content into the proper site area

Once we create the content, we assign it to a proper site area in order to view our page when we click the proper option in the navigator. To do this, follow these steps:

1. In the LWWCM interface, click **Site Management -> Site Framework**.
2. From the right-hand side, expand the list to **Intranet -> Home -> Insurance Products -> Personal and Families**.
3. Select **Life Insurance** and click **Edit**.
4. Click the **Properties** tab.
5. Click **Default Content**.
6. Select the **Life Insurance** content page.

7. Click **OK**, then **Save**, and then **Close**.

Now, let us check the results in the IT Jumpstart Web site. From the left main navigator, click the proper Insurance option. The recently created page appears with the content and images we selected. Figure 4-17 shows an example of the Life Insurance Web page.

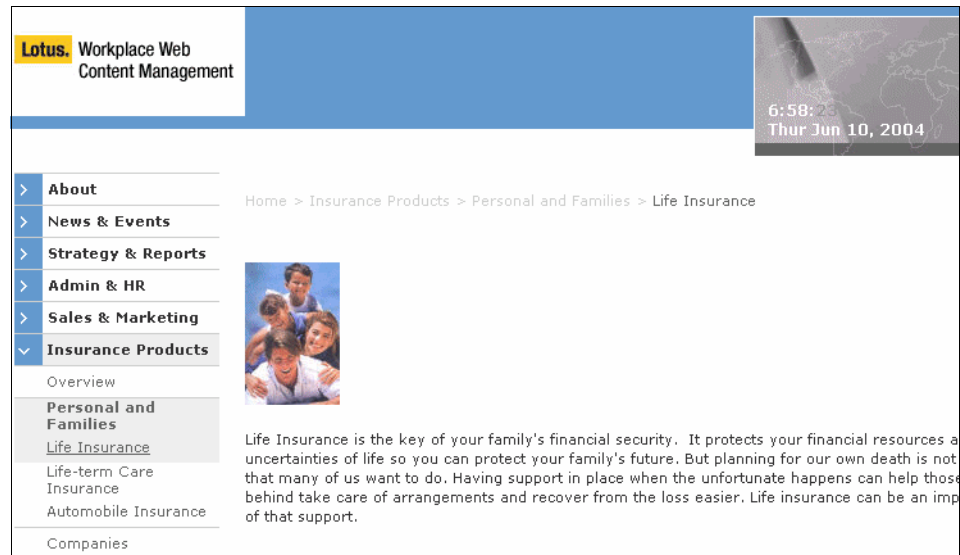


Figure 4-17 Life insurance Web page with navigator in the left hand side

4.4 Retrieving data from external sources into insurance content

LWWCM allows the retrieval of data from external sources and their publication to the Web site. This means that we can add information from other sources into our content and it looks to be part of the same content. Adding data from external sources into our content can be achieved in several ways:

- ▶ Federated content components
- ▶ JSP components
- ▶ Connect tags

In Chapter 7, “Publishing Content Manager content” on page 115, we show you how to retrieve data from the external sources to this case study Web site using the options listed above.

4.5 Troubleshooting

Our case study is based on the IT Jumpstart Web site sample. If you use syndication to create your new Web site, it may be possible that after syndication or modification to the existing Web site, some orphan site areas appear, or the site structure might get corrupt.

To fix this problem, try the following:


1. Stop the LWWCM server
2. Delete the content under the `app/ilwwcm/system/indexes` directory.
3. Restart the LWWCM server.



Part 2

Integration

In this part, before you begin integration, we cover various areas with which you need to be familiar and have set up to achieve a successful integration. We show you how to use Content Manager as a repository for LWWCM, and how to publish existing Content Manager content to LWWCM using federated content components, connect tags, and JSP components. With the basics of how to integrate, we address some of the issues you need to consider when planning the entire integration system architecture. This includes covering different scenarios of having LWWCM and Content Manager on one machine, having multiple LWWCMs using syndication, and having multiple Resource Managers in the setup. The part concludes with a discussion on portal integration.



Before you begin integration: pre-integration validation

Before you begin to integrate IBM Lotus Workplace Web Content Management (LWWCM) and IBM DB2 Content Manager (CM), you need to ensure you have the proper prerequisite software, that your LWWCM server is up and running, and that your Content Manager system is up and running.

Specifically, we cover the following topics in this chapter:

- ▶ Introduction
- ▶ Software requirements and case study setup
- ▶ Validation checklist
- ▶ Problem determination and troubleshooting

5.1 Introduction

Before you begin to work on the integration of LWWCM and Content Manager, you should be familiar with both products. We recommend reading the online documentation provided by both products to understand the logging and tracing functions build into LWWCM and Content Manager and understand various configuration files and their usage. We also discuss some network considerations.

5.1.1 Important log files and configuration files

Below is a summary of the most important log files for both the LWWCM and Content Manager products:

- ▶ LWWCM log files: in the <LWWCM_HOME>\connect\log directory.
<LWWCM_HOME> is where you installed LWWCM software.
- ▶ WebSphere log files: in the <WAS_HOME>\AppServer\logs directory.
<WAS_HOME> is where you installed WebSphere Application Server.
- ▶ IBM HTTP Server logs: in <IBM HTTP Server install directory>\logs directory.
It includes log files such as error.log, access.log, and admin_error.log.
- ▶ Content Manager Installation log files:
 - <ICMROOT>/logs/cm81install.log: logs program installation messages.
 - <ICMROOT>/config/icmcrldb.log: logs Library Server database creation process.
 - <ICMROOT>/config/icmcrmdb.log: logs Resource Manager database creation process.
 - <ICMROOT>/logs/database.log: created when access modules cannot be generated.

On AIX®, <ICMROOT> is on /usr/lpp/icm.

- ▶ The Library Server log file: icmsvr.log file.
Its location is specified in the Content Manager System Administration Client, Library Server configuration panel. By default, it is located:
 - On Windows, in C:\
 - On AIX, in /tmp
- ▶ Content Manager System Administration Client log file: in
<CMCOMMON>\ADMIN\COMMON\dklog.log
- ▶ Content Manager Windows Client tracing log file: ICMClient.log and ICMClient.err. Located in: C:\Documents and Settings\xxx\Application Data\IBM\Content Manager\log, where xxx is the Windows user ID used to

install Content Manager. Enabled through the ICMClientLog.ini file, which is in the same directory.

- ▶ VideoCharger log files: in <VC_HOME>\logs.
<VC_HOME> is where you installed VideoCharger software.

5.1.2 Some network considerations

There are some network considerations to take into account when you install LWWCM and Content Manager software. The network TC/PIP addresses of the server machines where you install the LWWCM and Content Manager software should always be static. Your short hostname should not contain special characters such as for example "? , _ ! , * " and others. If this is not possible, you should only use short host names during installation. In addition, you need to remove the DNS suffix from the machines with dynamic TCP/IP addresses. If there are still network problems, you should check to see if there is a port conflict. If so, you need to set a new number for the conflicted port.

Removing DNS suffix for machines with dynamic TCP/IP addresses

For dynamic (non-static) network TCP/IP addresses, make sure the DNS suffix is empty:

1. Select **My Computer -> Services -> Network Identification -> Properties -> More.**
2. Delete the DNS suffix entry in the Primary DNS suffix of this computer entry field.

Checking for port conflict

If you still have network problems, check if one or more of the communication protocol ports used by the LWWCM and Content Manager system have already been used from other programs:

- ▶ For AIX, you can run:

```
netstat -an | grep LISTEN or  
cat /etc/services
```
- ▶ For Windows, check the services file:

```
type C:\WINNT\system32\drivers\etc\services
```

Setting a different port

As an example, if your default IBM HTTP Server port 80 is already being used and you need to specify another port, follow the steps below:

1. In the IBM HTTP Server configuration file httpd.conf, change the line Port 80 to Port <new port number>.

2. Specify the new port in WebSphere Application Server Administration Console in the Host Alias List Box.
3. On the Content Manager System Administration Client window, route to the Resource Manager (Resource Manager) definitions and modify the Access Types List Box: Entry for http and change the port to the new port number.
4. Restart the IBM HTTP Server. Afterwards, restart the Resource Manager Web application (icrmr) using the WebSphere Application Server Administration Console or through Windows services if you set icrmr as a Windows service.

Problems starting WebSphere Application Server

If you have problems starting the WebSphere Application Server, you can add an entry combination of "TCP/IP < - > host name" to the host file on NT. The host is located in:

- ▶ For Windows: C:\WINNT\system32\drivers\etc\hosts
- ▶ For AIX: /etc/hosts

For example, add the following line in the hosts file:

```
9.1.38.11 demo1
```

If your TCP/IP address is dynamic, it may change when you reboot the machine. In this case, *do not forget* to change the TCP/IP-host name entry in the hosts file.

Socket error

If you get any socket errors on AIX, always contact your company network expert. Be sure that you have installed the right JDBC driver used by the WebSphere Application Server to access the Content Manager Resource Manager databases.

If your client connection to a server is refused by DB2, check if the cause is an invalid IP port:

- ▶ Check to see if the DB2 environment variable DB2COMM=TCPIP.
- ▶ Is the DBM CFG's SVCENAME set to the right instance's port number or service name? Check the *services* file if the name corresponds to an unused port.
- ▶ From the client side, check if the port is opened by executing the following in the DOS prompt:

```
telnet <host name> <port>
```

If this fails, then the error is not related to DB2.

5.2 Software requirements and case study setup

In this section, we list all the software requirements for the LWWCM server and the Content Manager server. Depending on the configuration you want to implement, you might have different ways of combining what software should be installed on which machines. For a more specific configuration set, refer to Chapter 8, "System architecture considerations" on page '167'.

5.2.1 Software requirements for LWWCM

LWWCM supports many operating systems and different back-end data repositories.

Operating systems

Operating systems that support LWWCM include the following with their required software levels:

- ▶ AIX 5.1 ML4, 5.2
- ▶ Solaris 8, Solaris 9
- ▶ Windows 2000 Server SP4
- ▶ Windows 2003 Server
- ▶ SuSE SLES 8.0 (Kernel 2.4) / Intel®
- ▶ Red Hat AS 2.1 (Kernel 2.4) / Intel
- ▶ OS400 V5R2 / iSeries™
- ▶ SuSE SLES 8.0 (Kernel 2.4) / zLinux

Application server

LWWCM supports the following application server with the required software level and fixpacks:

- ▶ WebSphere Application Server 5.0.1 or 5.0.2. The following fixpacks are mandatory: PQ75699, PQ76313, PQ77142, PQ80288, and PQ81144.

Data repository server

LWWCM supports the following system, database, and software (with the supported software levels) as its back-end data repository:

- ▶ File system
- ▶ DB2 7.2.x, 8.1 FP4
- ▶ DB2 400
- ▶ Oracle 9i (9.0.2.1)
- ▶ IBM Content Manager 8.2
- ▶ Cloudscape™ 5.1.26
- ▶ SQL Server 2000 SP3a
- ▶ Informix® 9.4

5.2.2 Software requirements for Content Manager

We list the software requirements for Content Manager V8 for Multiplatforms, including those for Windows and for AIX.

Windows

Running Content Manager V8.2 on Windows NT® or Windows 2000 requires the following software and software levels (Always see the Content Manager installation README information (<ICMROOT>\readme) for the latest version requirements of prerequisite software, including related update or fixpack.):

- ▶ Database for both Library Server and Resource Manager
 - DB2 Application Development Client (known as DB2 Software Development Kit, or SDK) and one of the following:
 - DB2 Universal Database™ (DB2 UDB) Enterprise Edition version 7.2 (or later)
 - IBM DB2 UDB Enterprise Extended Edition version 7.2.1 with latest fix pack
 - IBM DB2 UDB Enterprise Server Edition version 8.1 (included in the IBM Content Manager package with a limited use license)
 - For optional support of Oracle:
 - Oracle 8.1.7.4 or later *or* Oracle 9.2.0.1 or later
- ▶ Library Server:
 - Microsoft Visual C++ 6.0 or Microsoft Visual Studio.Net Professional
 - IBM DB2 Information Integrator Relational Wrappers version 8.1 (included in the IBM Content Manager package with a limited use license)

Note: Use of IBM DB2 Information Integrator Relational Wrappers version 8.1 requires IBM DB2 UDB Enterprise Server Edition version 8.1 (included in the Content Manager package with a limited use license).

- ▶ Resource Manager:
 - WebSphere Application Server V4.0.3 or later or WebSphere Application Server 5.0 (included in IBM Content Manager package with a limited use license)
 - Tivoli Storage Manager API Client V4.2.1 (or later), and Tivoli Storage Manager Server version 4.2.1 (or later) if you want to provide long-term storage for your objects on devices other than the fixed disks

- ▶ For text search:
 - Text Information Extender V7.2 (if using IBM DB2 UDB version 7.2)
 - IBM Net Search Extender V8.1 (if using IBM DB2 UDB version 8.1)

AIX

Running Content Manager V8.2 on AIX requires the following software and the software levels:

- ▶ AIX 4.3.3 with fix pack 8, or later versions of AIX
- ▶ TCP/IP (installed with AIX)
- ▶ Database for both Library Server and Resource Manager
 - DB2 SDK for AIX and one of the following:
 - IBM DB2 UDB Enterprise Edition 7.2 (or latter)
 - IBM DB2 UDB Extended Enterprise Edition 7.2 with the latest fix pack
 - IBM DB2 UDB Enterprise Server Edition V8.1 (included in the IBM Content Manager package with a limited use license)
 - For optional support of Oracle:
 - Oracle 8.1.7.4 or later, or
 - Oracle 9.2.0.1 or later
- ▶ Library Server compiler:
 - VisualAge® C++ Professional batch compiler (for AIX 4.3.3, V5.0.2.0)
- ▶ IBM DB2 Information Integrator Relational Wrappers V8.1 (included in the IBM Content Manager package with a limited use license)

Note: Use of IBM DB2 Information Integrator Relational Wrappers V8.1 requires IBM DB 2 UDB Enterprise Server Edition V8.1 (included in the Content Manager package with a limited use license).

- ▶ Resource Manager:
 - WebSphere Application Server V4.0.3 or later, or
 - WebSphere Application Server 5.0 (included in IBM Content Manager package with a limited use license)
 - Optional: Tivoli Storage Manager API Client V4.2.1 (or later) and Tivoli Storage Manager Server V4.2.1 (or later) if you want to provide long term storage for your objects on devices other than the fixed disks.

- ▶ For text search:
 - Text Information Extender V7.2 (if using IBM DB2 UDB version 7.2) or
 - IBM Net Search Extender V8.1 (if using IBM DB2 UDB version 8.1)

5.2.3 Case study setup

Here, we list the software requirements for the case study we set up for this redbook.

Windows

For our case study setup on Windows, we use the following software and the software levels:

- ▶ IBM WebSphere Application Server V5.0.2.2 with IBM HTTP Server 1.3.26.
- ▶ IBM Lotus Workplace Web Content Manager V2.0 (LWWCM).
- ▶ Microsoft Visual C++ V6.0.
- ▶ DB2 Enterprise Server Edition V8.1 (DB2 ESE).
- ▶ DB2 NetSearch Extender V81 (DB2 NSE) and Fixpack IC35476.
- ▶ IBM Content Manager V8.2, Fixpack 3, enable UTF-8 during Library Server Installation, including Content Manager System Administration Client.
- ▶ IBM Content Manager V8.2 Windows Client and/or eClient to import *Document* item objects into the Content Manager system, see “Data modeling” on page 25.
- ▶ IBM Information Integrator for Content V8.2, Fixpack 3, EIP Server with Administration Client, Content Manager V8 and the Federated connector. They are used to create federated search templates and to create new federated content Components in LWWCM (see 7.3, “Federated content component” on page 123).
- ▶ IBM DB2 Content Manager VideoCharger Server and Player V 8.2.
- ▶ Internet Explorer Browser, the Java Runtime Environment 1.3.9 for LWWCM server is installed automatically on first time startup of the LWWCM user interface. Prerequisite is an *existing* Internet connection. If you are using Internet Explorer as the Web browser, you should also update to V6 including SP1 or higher.

AIX

For our case study setup on AIX, we use the following software and the software levels:

- ▶ Java 1.3.1.
- ▶ IBM Visual Age C++ batch Compiler, V5.1.0.25.
- ▶ IBM WebSphere Application Server V5.0 and Fixpack 5.0.1 with IBM HTTP Server 1.3.26.
- ▶ IBM Lotus Workplace Web Content Manager V2.0 (LWWCM).
- ▶ DB2 Enterprise Server Edition V8.1.1.4 (DB2 ESE).
- ▶ DB2 NetSearch Extender V81 (DB2 NSE) and Fixpack IC35476.
- ▶ IBM Content Manager V8.2, Fixpack 3, enable UTF-8 during Library Server Installation.
- ▶ IBM Information Integrator for Content V8.2, with Information Integrator for Content federated (FED) and Content Manager (ICM) connectors.
- ▶ Netscape Explorer, the Java Runtime Environment 1.3.9 for LWWCM server is installed automatically on first time startup of the LWWCM user interface. Prerequisite is an *existing* Internet connection.

5.2.4 How to check installed software versions

For some of the required software, you can check the version as follows:

- ▶ For DB2, from a DB2 command window, execute **db2level**.
- ▶ For WebSphere, run the versionInfo.bat file.
- ▶ For IBM HTTP Server, view the version.signature file in the IBM HTTP Server install directory.
- ▶ For LWWCM, see the version information in the LWWCM user interface.
- ▶ For IBM Content Manager, enter **cmlevel.exe** in the <ICMROOT>\bin or enter **cmlevel** on AIX.
- ▶ For Information Integrator for Content, execute **cmlevel.exe** (AIX **cmlevel**) in the <CMBROOT> directory (AIX <CMBROOT>/bin)

For advanced users, you can also open on Windows Command Prompt window, enter **regedit** and select **My Computer -> HKEY_LOCAL_MACHINE -> Software**. From there, you can find your installed IBM software, open the tree view and click the Current Version folder. On AIX, you can use **smitty** and search for example for the Information Integrator for Content product to see the installed version for the content connectors.

5.3 Validation checklist

Before you start integration of LWWCM and Content Manager, you should go through the validation checklist to ensure that everything is installed properly. If you get any error on one of the validation steps, refer to 5.4, “Problem determination and troubleshooting” on page 78.

5.3.1 WebSphere Application Server and IBM HTTP Server

WebSphere Application Server and IBM HTTP Server are essential in Content Manager and LWWCM solutions. We need to verify that it is installed correctly.

Verify the IBM HTTP Server installation

To verify that the IBM HTTP Server is installed correctly, do the following:

1. Start the IBM HTTP Server in the Windows services control panel.
2. Open a browser window and enter:

```
http://<your host name>/snoop
```

Verify the WebSphere Application Server installation

To verify that the WebSphere Application Server is set up and runs properly, do the following:

1. Check to see if it is running; open a Web browser and enter:

```
http://<hostname>:<port>/snoop
```

Where <hostname> is the domain name or IP address of your application server and <port> is the port number of your WebSphere Application Server (by default, this is 9080) server.

2. Make sure you are using WebSphere Application Server V5.0.1 or 5.0.2, with the mandatory fixpacks: PQ75699, PQ76313, PQ77142, PQ80288, and PQ81144. If you install V5.0.2.2, the fixpacks are automatically included.

To see which version you are using on your system:

- a. Open the WebSphere Application Server Administration console: select **Start -> IBM WebSphere -> Administrative Console**.
- b. On the first page (as shown in Figure 5-1 on page 73), you can see the WebSphere Application Server version. There is also build number and build date information on it.

Alternatively, you can go to the file system where you install the WebSphere Application Server to see the version information:

- a. Go to <install_root>\WebSphere\AppServer\properties\version.

- b. Open the file BASE.product to see the current version.

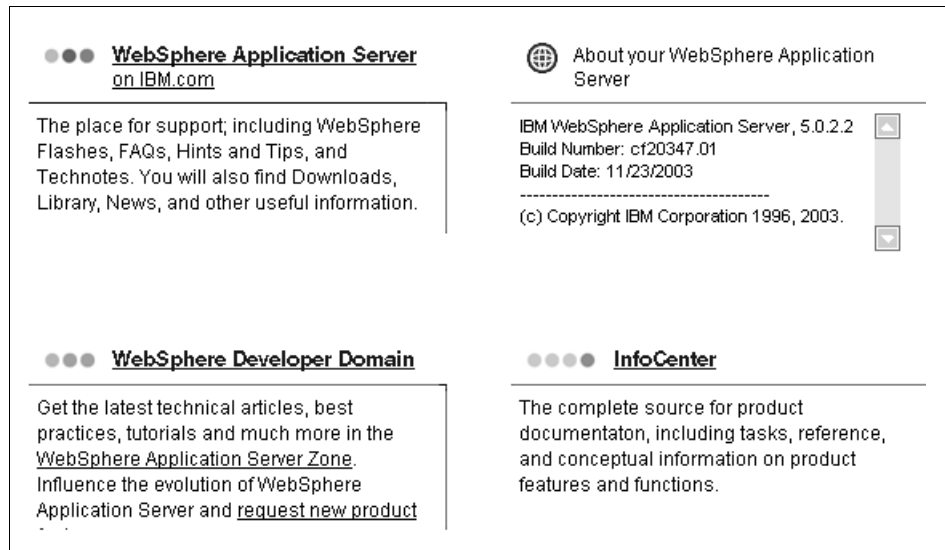


Figure 5-1 WebSphere Application Server administration console

5.3.2 LWWCM

WebSphere Application Server can host more than one application server. You can install your LWWCM Web application on *server1* or (optionally) define a new application server.

If you define a new application server, and name it, for example, LWWCM, to host your LWWCM Web application (LWWCM.war), you need to start the server manually:

1. Open a command prompt.
2. Change the directory to: <installation_root>\WebSphere\AppServer\bin.
3. Enter startServer.bat LWWCM.

Alternatively, WebSphere Application Server has a tool to register WebSphere Application Server as a Windows service so you do not have to start the server manually. To set it as a Windows service:

1. Open a command prompt.
2. Change the directory to <installation_root>\WebSphere\AppServer\bin.
3. Enter wasservice -add "LWWCM Server" -servername LWWCM -userid <wasadmin> -password <password>

where LWWCM Server is the name that will appear as a Windows service, and LWWCM is the application server name you created.

Verify the LWWCM installation

To verify that the LWWCM installation is successful, do the following:

1. Open a Web browser and enter:

```
http://<hostname>:<port>/<context_root>/connect/?MOD=AJPEGUI
```

This command builds the LWWCM user interface where:

- <hostname> is the domain name or IP address of your application server.
- <port> is the port number of your application server where the LWWCM Web application is run.
- <context_root> is the context root you enter when you install LWWCM on WebSphere Application Server.

2. From the Web browser, enter:

```
http://<hostname>:<port>/<context_root>/connect/aprix
```

Enter Administrator for the user name, and password for the password to access the LWWCM user interface. These are the default user name and password for the LWWCM user interface. JRE 1.3.9 needs to be installed. If you are connected to the Internet, you will be prompted to install the JRE plug-in.

Important: The JIT compiler setting needs to be specified to stop the JIT performing incorrect optimizations of LWWCM code. Since the JIT compiler takes into account many factors before optimizing various parts of the code, it is possible to run a system for days before experiencing issues. It is thus recommended to set this setting on all servers.

See the *LWWCM Information Center Installation Guide* for the correct setting.

5.3.3 DB2 and NSE

Verify your DB2 and NSE server installation by:

1. Open DB2 command window, and enter db2start to start the database manager on your system.
2. Select **Start -> Programs -> IBM DB2 -> Set-up Tools -> First Steps** and click **Create Sample Database**.
3. From the DB2 command window, enter the following commands to see a list of databases defined on the system, to check the sample database connection, and to see a list of tables in the sample database:

```
db2 list database directory
db2 connect to Sample
db2 list tables
```

4. From the DB2 command window, start the Net Search Extender by entering the following command:

```
db2text start
```

You can also set the NSE service *DB2EXT - DB2* to automatically start in the Windows services control panel.

To access a remote DB2 database, you can use the DB2 command window, the DB2 Command Center or DB2 Control Center. Refer to 6.2, “Accessing a remote Library Server database” on page 97 for more information.

5.3.4 IBM Content Manager and Information Integration for Content

The following is a checklist to validate that Content Manager and Information Integrator for Content are installed properly:

- ▶ Verify the installation log files in: <ICMROOT\logs\>, where <ICMROOT> is the Content Manager product install location. Make sure there are no error messages in the log files.

- ▶ Verify the Library Server database connection by enter the following command in a DB2 command window:

```
db2 connect to <LSDB> user <LSadminID> using <password>
```

where <LSDB> stands for the Library Server database name, <LSadminID> stands for Content Manager system administrator ID, and <password> stands for the password for the system administrator ID.

- ▶ Validate that the Library Server database tables exist by enter the following command in a DB2 command window:

```
db2 list tables
```

- ▶ Verify that the Library Server access modules for the Content Manager item types are created properly. The Library Server uses stored procedure calls to access its database. Look for DLL files in the following directory:

```
<ICMROOT>\<LSDB>\d11
```

If you find files ending with *.tx1, *.tx2,, *.tx3 , *.tx4, refer to 5.4.4, “IBM Content Manager and Information Integration for Content” on page 83 to resolve the problem.

- ▶ Verify the Resource Manager database connection by entering the following command in a DB2 command window:

```
db2 connect to <RMDB> user <RAdminID> using <password>
```

Where <RMDB> stands for the Resource Manager database name, <RMAadminID> stands for Resource Manager administrator ID, and <password> stands for the password for the administrator ID.

- ▶ Start the WebSphere Application Server Resource Manager application server.

We recommend adding the Resource Manager application as a WebSphere application service by entering the following command at the command prompt in the WebSphere Application Server bin directory:

```
wasservice -add "Resource Manager" -servername <RM_Web_App_Name> -userid <RMAadminid> -password <password>
```

We also recommend changing the Startup type for the Resource Manager application service to Automatic in the Windows Services control panel.

- ▶ To test the deployment of the Resource Manager servlets, open your Web browser and enter the following URL:

```
http://<hostname>/<RM_Web_App_Name>/ICMResourceManager
```

Where <hostname> stands for your fully qualified server name.

- ▶ For secure communication, Secured Sockets Layer (SSL) is used. SSL is *required* for performing Resource Manager configuration from your Content Manager System Administration Client. To test your SSL configuration, enter the following URL in your Web browser:

```
https://<hostname>/<RM_Web_App_Name>/ICMResourceManager
```

If the response is the following, then your Resource Manager Web application is up and running:

```
IBM Content Manager - null - No order found to process OK
```

If this is working, you can start to define and configure Resource Manager storage systems such as TSM, file system or VideoCharger with the Content Manager System Administration Client. Read the Content Manager installation manual for the SSL setup details. Below, we provide a very short description of the necessary configurations so that the *https* URL to the Resource Manager Web application will deliver the described response:

- a. Create the Secure Socket Layer (SSL) key by clicking **Start -> Programs -> IBM HTTP Server 1.3.26 -> Start Key Management Utility**.
 - b. Activate the IBM HTTP Server for SSL communication by adding the necessary information to the httpd.conf file.
- ▶ Start and run the Content Manager System Administration Client to make sure everything works:
 - a. Select **Start -> Programs -> IBM Content Manager for Multiplatforms V8.2 -> System Administration Client**.

- b. Check that you can access Library Server configurations.
- c. Check that you can access Resource Manager configuration.
- d. Check that you can create Resource Manager System Managed Storage (SMS) objects.

Refer to 2.3, “Content Manager architecture overview” on page 22 for more information on Library Server, Resource Manager, and System Administration Client.

- Verify that the Content Manager (ICM) connector and the federated (FED) connector or any other connectors are installed properly. The configuration settings are stored in the following files:
 - cmbicmsrvs.ini: contains a list of available Library Servers.
 - cmbds.ini: contains a list of available Information Integrator for Content servers.

If you want to add remote Library Servers or Information Integrator for Content servers, update the above files appropriately.

To test the ICM and FED connections, use TConnect Java bean. This bean is located in your Information Integrator for Content Information Center subdirectory:

<CMCOMMON>\infoctr\info\apiinfo\dk\samples\java\beans

The syntax for this Java bean is:

```
java TConnect <dstype> <server> <userid> <password>
```

Where <dstype> is the type of back-end server to connect. Valid values include: Fed, ICM, DL, OD, and DB2.

As an example, in our scenario, we run the commands:

```
java TConnect ICM icmnlbdb icmadmin password
java TConnect Fed cmbdb cmbadmin password
```

- Verify that Content Manager server works properly by load sample data and perform search and import functions:
 - a. Select **Start -> Programs -> IBM Content Manager for Multiplatforms V8.2 -> First Steps** and click **Load Sample Data**.
 - b. Launch Windows Client: select **Start -> Programs -> IBM Content Manager V8 -> Client for Windows for Multiplatforms V8.2**.
 - c. Log in. Be familiar with the client interface. Perform some search and import functions within the Windows Client. Make sure these functions work.
- If you installed eClient, validate that it is up and running properly by entering the following URL in a Web browser:

`http://<hostname>/eClient82/IDMInit`

To validate:

- a. Log in to any local or remote Library Server from the eClient.
- b. Perform a search on any existing documents.
- c. View the defined Worklists.
- d. Import some new documents.
- e. In general, test the different functions and features which you have enabled in your eClient configuration file *idm.properties*. If some of the functions are not available, you can modify the file to make the features available.

We recommend setting the eClient application as a WebSphere application service in the Windows services control panel. We find it easier to start and stop the application this way.

5.4 Problem determination and troubleshooting

In this section, we try to address some of the typical problems you might encounter while installing and configuring WebSphere Application Server, DB2, Net Search Extender, Content Manager, Information Integrator for Content, and LWWCM. We provide some guidance on problem determination, troubleshooting and hints and tips. These troubleshooting information are based on the experiences from worldwide Content Manager certification classes.

5.4.1 General hints and tips

Regardless of how your system configuration is set up, the clients and the servers in your LWWCM and Content Manager solution should, if possible, *maintain the same release and modification level* for drivers and products. If you have different product releases and modifications levels, read the product installation and prerequisite software documentation about recommended packages, fixpack levels and operating system versions. Ensure that your system works properly and avoid the unnecessary effort later on to bring everything up and running.

Try to install on *clean systems* for the LWWCM and Content Manager server installation. If you installed a previous version of software on the machine, the machine may still have some files on the system and the previous setup may still remain on the system. This could cause problems during installation and make the troubleshooting process unnecessarily involved. In addition, for performance and maintenance reasons, starting with clean systems always make sense.

If you plan to migrate from previous versions, *always read the migration documentation.*

Prepare a clean system for installation

Below, we describe, as an example, how you can guarantee that you install the products on a clean Windows system for a new LWWCM and Content Manager system setup:

1. Remove previous versions if no longer needed: select **Start -> Settings -> Control Panel -> Add/Remove Programs**. Always reboot after a full uninstallation.
2. For experienced Windows user, use **regedit** to clean up any remaining system setup and configuration that was not cleaned from the the previous step:
 - a. From a command prompt window, enter `regedit`.
 - b. Select **My Computer -> HKEY_LOCAL_MACHINE -> Software**. Remove any existing entries where the programs have already been removed.
 - c. Delete any corresponding services entry for these programs in:
 - **My Computer -> HKEY_LOCAL_MACHINE -> System -> ControlSet00.. -> Services**
 - **My Computer -> HKEY_LOCAL_MACHINE -> System -> CurrentControlSet -> Services**

Note: Be very careful when you perform this task. If you are not familiar with **regedit**, we do not recommend you doing this.

3. Delete any leftover subdirectories from previous software installations.

5.4.2 WebSphere Application Server

You can find the log files for WebSphere Application Server under `<install_root>\WebSphere\AppServer\logs`.

JDBC driver

The Resource Manager Web application (which is installed on WebSphere) needs to be able to connect the Resource Manager database (RMDB) using the JDBC driver. The database connections will fail if the `db2java.zip` file is not found in the WebSphere Application Server classpath. If the file is not found in the classpath, the WebSphere Application Server standard error log file will contain a message indicating that the DB2 JDBC driver could not be found.

To verify that the classpath in WebSphere Application Server:

1. Open the WebSphere Application Server Administrative Console: select **Start -> Programs -> IBM WebSphere -> Application Server v5.0 -> Administrative Console**.
2. Go to **Servers -> Application Server -> icrm -> Process Definition -> Java Virtual Machine** where **icrm** is the Resource Manager server.
3. Add the full path of the db2java.zip file in the classpath field.

Access the directory

The owner of the WebSphere processes should have the permissions to access and create files within the LWWCM directories because the LWWCM engine has to create some necessary files to complete the initialization process. Make sure the permissions are set correctly.

5.4.3 DB2 and NSE

In this section, we present some hints and tips, and common problems related to DB2 and NSE.

NSE

For NSE, you should avoid running *NSE index text document updates* too frequently (for example, every minute) because this might have a negative effect on search performance, indexing performance, and might result in locking conflicts. The update index cycles can be specified per item type on the Text Search Option panel in the Content Manager System Administration Client or with C++ or Java applications using the NSE APIs.

Some other hints and tips for NSE are:

- ▶ Ensure that there is enough space on the server. Approximate four times the size of the Content Manager document text data you want to enable for text search.
- ▶ Specify, in the Content Manager Administration Text Search Option panel, the different index directories for different text search enabled item types.
- ▶ Specify different index and work directories.
- ▶ To get trace informations on NSE, use the DB2 trace file db2trc:

```
db2trc on -f db2trc.dmp
```

Search for NSE functions using the mask “*.96.*” in the db2trc.dmp file.

A common NSE problem and its solution are described below:

- **Problem:** You create a new item type, mark it text searchable, and on the text search option panel, under the User Defined Function (UDF) name, you select **ICMfetchFilter**. The maximum size of the objects that can be processed is limited to 8 MB. In your environment, you need to handle files with sizes greater than that.

Solution: To handle text document objects up to 25 MB, you need to perform the DB2 commands shown in Example 5-1 in a DB2 command window.

Example 5-1 DB2 commands to handle text document objects up to 25 MB

```
Connect to database: db2 connect to <dbname> user <icmadminUser> using
<password>
drop function ICMfetchFilter;
create function ICMfetchFilter ( VARCHAR(512) ) RETURNS CLOB(25M) EXTERNAL NAME
'ICMNLUF!ICMfetch_Filter' LANGUAGE C PARAMETER STYLE DB2SQL FENCED NO SQL;
drop function ICMfilter;
create function ICMfilter ( CLOB(25M) ) RETURNS CLOB(25M) EXTERNAL NAME
'ICMNLUF!ICM_filter' LANGUAGE C PARAMETER STYLE DB2SQL FENCED NO SQL;
db2 connect reset
```

Note that, when setting UDF, you have two choices:

- ICMfetchContent: used to enable text search on the default supported types, such as HTML, XML, and GPP.
- ICMfetchFilter: used to enable text search for document types other than the default supported types.

DB2

To catalog a remote Library Server database (for example, ICMNLSDB), using **db2cmd** or the DB2 Control Center Assistant, you must know the remote server host name, the database name, and the database instance port number. You also should define alias names for the different remote databases to know which database you are working on if they all have the same local Library Server database name. For more information, refer to 6.2, “Accessing a remote Library Server database” on page 97.

You can also find DB2 troubleshooting hints and tips at the following Web site:

<http://www-306.ibm.com/software/data/support/>

Some information you should be aware of for DB2 troubleshooting (and performance improvement) includes:

- The DB2 trace utility and log file:
 - Trace utility: <db2 install directory>\bin\db2trc.exe
 - Log file: <db2 install directory>\DB2\db2diag.log

- ▶ Some of the most important performance-relevant DB2 parameters and practices on your LWWCM and Content Manager installation are:
 - DB2 buffer pool: Update to no more than 50% of physical memory.
 - Sort heap
 - Number of agents
 - maxappls: Maximum number of concurrent application connections
 - maxagents: Set at least as high as the sum of maxappls
 - locklist: May need to be increased if there is a large number of concurrent users
 - Number of nodes: Performance scales linearly with number of nodes used.
 - Place the DB2 database log on a separate physical disk from the Library Server database.
- ▶ If you are not familiar with performance-related configuration parameters, use the configure Performance Wizard to tune your Library Server and Resource Manager databases. Change only one parameter at a time to isolate the changes and minimize undesired results.
- ▶ Back up your Library Server and Resource Manager databases.
- ▶ When your performance starts to decrease as you access, search or import objects in Content Manager, run the following commands from the DB2 command window:

```
db2 reorgchk
update db2 runstats
db2rbind
```

The last command updates the application's statistics access plan to access the Library Server and Resource Manager database. See the DB2 online documentation for details on how to use and run these commands.

A common DB2 problem and its solution are described below:

- ▶ **Problem:** You get the error shown in Example 5-2.

Example 5-2 TCP/IP sockets error

```
SQL30081N  A communication error has been detected.  Communication protocol
being used: "TCP/IP".  Communication API being used: "SOCKETS".  Location where
the error was detected: "".  Communication function detecting the error:
"connect".  Protocol specific error code(s): "10061", "*", "*".SQLSTATE=08001
```

Solution: In our setup, the cause of this error was the use of a setup security policy tool Zone Labs Integrity Desktop with firewall protection and program control.

To solve the problem, open the Zone Labs Integrity Desktop or whatever security tool you are using. Go to Program Control and give all rights to the IBM DB2 programs/executables you can find. See Figure 5-2 as an example.

Figure 5-2 Zone Labs Integrity desktop

Be familiar with the online Information Center for both Content Manager and Information Integrator for Content. They contain some troubleshooting guidelines of use during product installation.

The following Web sites also provide support for both products:

Ensure sufficient paging space. We recommend increasing the paging database size to 1 GB or more. You can do this in Windows: select **Start-> Setting->**

Control Panel -> System -> Tab Advanced-> Performance Options -> Change.

For Resource Manager, the default storage is the file system. You should ensure that at least one drive is NTFS formatted and labeled.

The following is a list of problems and their solutions that may help you during your Content Manager and Information Integrator for Content installation.

- **Problem:** You create new item types with the Content Manager System Administration Client. After refreshing the view, you cannot see the new item type, or you get error messages stating that there are no access modules or item types.

Solution: Check the <ICMROOT>\<LSDB>\DLL subdirectory where <ICMROOT> is the Content Manager product install directory, and <LSDB> is the Library Server name. You should find the DLL files representing access modules which are used for Content Manager item types. These modules are generated dynamically during the installation using the C++ compiler. If you find files ending with:

- *.tx: This implies you have a precompile error.
- *.tx2: This implies you have a compile error.
- *.tx3: This implies you have a link error.
- *.tx4: This implies you have a build error.

Verify:

- That you have *moved* and not copied the Microsoft C++ environment variables from USER variables to SYSTEM variables as described in the *Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332. Delete any empty PATH, LIB and INCLUDE environment settings from the USER variables.
- From the command prompt, that your PATH, LIB, INCLUDE environment settings are not broken in any way. For example, you may have a comma instead of semicolon. If so, go back and correct them.
- Check that you really did register the Microsoft C++ environment variables during the compiler installation, for example with **regedit**. To see the Microsoft C++ environment variables, you must set open the file vcvars32.bat at the location where you installed the C++ compiler.

After you correct your Microsoft C++ environment variables settings, delete all files in the <ICMROOT>\<LSDB>\DLL subdirectory. Compile and run the **TRebuildCompTypeICM.class** utility in the <ICMROOT>\conf subdirectory, to rebuild the access modules.

- **Problem:** You receive a semaphore error when starting the Resource Manager application server (in our setup, it is called icmrm) from the Windows

services panel. You cannot store or retrieve Resource Manager objects from your own application, Content Manager Windows Client, or eClient.

Solution: Verify that your user ID and password to start the Windows service are correct. Try to start the Resource Manager Web application from the command line **startServer icmm**. If it is still not working after rebooting and your network connection is up and running, then the reason might be that the WebSphere Application Server was not started prior to the installation of Content Manager, which includes the installation of Resource Manager. Compile and run the WebSphere Application Server deploy utility **WAS50DeployRM.class** in the <ICMROOT>\conf subdirectory.

- **Problem:** Logon to the Content Manager System Administration, Windows Client, or eClient fails.

Solution: Be sure that your password and logon user ID is correct. Update the <CMCOMMON>\cmbicmenv.ini file to make sure that it contains a valid database connect user ID and password. <CMCOMMON> is the shared area for Content Manager and Information Integrator for Content (previously known as Enterprise Information Portal) configuration files. The reason is that first the logon user is used to connect to the Library Server database. In our setup, it is ICMNLSDB. If this fails, the defined database connection user ID (in our setup. icmconct) as defined on the system and during Content Manager installation will be used to connect. Finally, it will check if the login user is a valid user defined in the Content Manager's Library Server database table, ICMUSER.

- **Problem:** You cannot open the Resource Manager configuration in the Content Manager System Administration Client. This means you cannot define and view System Managed Storage (SMS) objects such as collections.

Solution: Check that the Resource Manager ports, defined in the Resource Manager properties panel of the Content Manager System Administration Client, for the Resource Manager HTTP and secure socket layer HTTPS protocol are correct. Compare the settings with the WebSphere Application Server Administration Console Resource Manager (icmm) server transport entries.

Check the created key certificate for the HTTP server. Your entries may be invalid. Recreate it if necessary.

Check and correct any incorrect SSL configuration entries in the <HTTP_HOME>\conf\httpd.conf file where <HTTP_HOME> stands for HTTP server installation directory.

Example 5-3 on page 86 shows the SSL configuration part of the httpd.conf files from our Windows test machine akaokyrb. Example 5-4 on page 86 shows the SSL configuration part of the httpd.conf files from our AIX test machine zaire.

Example 5-3 SSL configuration in httpd.conf for Windows

```
LoadModule ibm_ssl_module modules/IBMModuleSSL128.dll

Listen 443
<VirtualHost akaokyrb.svl.ibm.com:443>
DocumentRoot "c:/IBM/ibmhttpserver/htdocs/en_us"
Keyfile "c:/IBM/ibmhttpserver/key.kdb"
SSLEnable
SSLV2Timeout 100
SSLV3Timeout 1000
SSLClientAuth none
SSLServerCert icmrn
SSLCipherSpec 64
SSLCipherSpec 62
SSLCipherSpec 3A
SSLCipherSpec 39
</VirtualHost>
```

Example 5-4 SSL configuration in httpd.conf for AIX

```
LoadModule ibm_ssl_module libexec/mod_ibm_ssl_128.so
<VirtualHost "zaire.svl.ibm.com:443">
DocumentRoot /usr/IBMHttpServer/htdocs/en_US
Keyfile /usr/IBMHttpServer/keys/key.kdb
SSLV2Timeout 100
SSLV3Timeout 1000
SSLEnable
SSLClientAuth none
SSLServerCert icmrn
SSLCipherSpec 64
SSLCipherSpec 62
SSLCipherSpec 3A
SSLCipherSpec 39
</VirtualHost>
```

- **Problem:** Since Content Manager V8.2, the Library Server has introduced a service process called the Library Server Monitor Process. This process checks on predefined intervals if the Resource Managers defined in Library Server are up and running. The purpose is to support replica failover. Resource Managers which are not responding are marked as *Not available*, and therefore cannot be accessed by other applications such as LWWCM to access or store objects. In a replication setup with failover environment, the Library Server will access another Resource Manager that has a replica copy of an object. If you do not use Content Manager V8.2 FP1, there is a bug such that the Library Server would work “perfectly” with a Resource Manager that was marked offline.

- **Solution:** If you do not use a replica failover setup environment, you should turn off this monitor process. On Windows, just disable this service. On AIX, run the command: `/etc/rc.cm1sproc -shutdown`. Better yet, just apply Content Manager V8.2 FP1. Also, you must manually mark your Resource Manager as Available again by updating a table in the Library Server database:

- a. Execute the DB2 query in the DB2 command window:

```
db2 connect to icm1sdb user icmadmin using password
db2 "select rmcode,rmname,rmflags from icmstresourcemgr"
```

- b. For each entry that you find where `rmflags>1`:

- If `rmflags` is an EVEN value, run:

```
db2 "update icmstresourcemgr set rmflags=0 where rmcode=<rmcode
listed from the previous query>"
```

- If `rmflags` is an ODD value, run:

```
db2 "update icmstresourcemgr set rmflags=1 where rmcode=<rmcode
listed from query>"
```

- c. Commit the changes:

```
db2 commit
db2 terminate
```

5.4.5 LWWCM

LWWCM write its log files in the directory `<LWWCM_HOME>\connect\log` where `<LWWCM_HOME>` is the Web Content Manager home folder where you copied the `/app` folder during the installation.

Configuring the log level

In the `connect.cfg` file, you can manage and configure the log files to specify the amount of information to be logged.

In Example 5-5, the `error.log` file is stored in the `<LWWCM_HOME>\connect\log` directory. When restarting LWWCM, the errors will be appended to the end of the existing error log file with trace date and time.

Example 5-5 Configuration for LWWCM logs

```
<!-- Log manager configuration -->
<LogManager>
  <ErrorLog>
    <File LogFile="../connect/log/error.log" FlushLog=false Buffered=true
TraceTime=true TraceDate=true TraceLevel=2 Rollover=Size MaxFileSize=5M />
  </ErrorLog>
  <FullLog>
```

```

        <File LogFile="../connect/log/connect.log" FlushLog=false Buffered=true
TraceTime=true TraceLevel=1 TraceDate=true TraceThread=true Rollover=Size
MaxFileSize=5M />
        <Screen Buffered=false TraceTime=true TraceLevel=0 TraceDate=false
TraceThread=true />
        </FullLog>
        <DebugLog>
        <File LogFile="../connect/log/debug.log" FlushLog=false Buffered=false
TraceTime=true TraceLevel=0 Rollover=Size MaxFileSize=5M>
        <Packages>
        <!-- by default not debugging any package -->
        </Packages>
        </File>
        </DebugLog>
</LogManager>

```

It is possible to delete the log files after stopping the application server. When you start the application server, it creates a new set of log files. This will give you an easy way to find problems during the initialization process of LWWCM Web application. Alternatively, you can set `FlushLog=true` in the `connect.cfg` file to create a new log file each time the LWWCM is started. For more information, refer to the online Information Center for LWWCM.

Syndicator and subscriber

In the `connect.cfg` file, you can set the default encoding for a LWWCM site. By default, it is `DefaultEncoding value="UTF8"`. If you plan to use more than one LWWCM server and you want to use syndicators and subscribers to synchronize them, then they must have the same default encoding value; otherwise, the content may be corrupted when syndication occurs.

The administrators' group will not be synchronized using the syndicators and subscribers documents.

When you create syndicator and subscriber objects with LWWCM, save the syndicator first before saving the associated subscriber document; otherwise, you will get an error. To create a replication, you may use either the subscriber document or the associated syndicator document. To pull the documents manually from the syndicator server:

1. Open the subscriber which you have already created in read mode.
2. Click **Subscriber Update** to update all changes that have been made since the last replication or click **Subscriber Rebuild** to rebuild all the documents from the syndicator.
3. Click **Refresh**, and check the result.

4. Check the current status of the subscriber. The syndicator and subscriber must have the same current status if they are synchronized.

It may take some time to syndicate (replicate) when dealing with a large Web site.

Using a search template

If you define a new search template within the Information Integrator for Content Administration Client to use with LWWCM, you might see it in LWWCM, but you will be not able to use it until you restart the LWWCM Web server.

Using version control

When you create and manage your Web site, we recommend that you use the versioning feature of LWWCM to save, read, restore and delete the object. At the time of writing, we do not recommend using the versioning feature of Content Manager when you work with objects created from the LWWCM user interface. You can use the versioning option of Content Manager if you want to access different object versions stored in Content Manager using the Content Manager clients (eClient, Windows Client, or integrated solution). In this case, you can use the connect tags, the federated content components, and the JSP components of LWWCM. Refer to 7.3, “Federated content component” on page 123, 7.4, “Connect tags” on page 139, and 7.5, “JSP component” on page 151 for more details.

Indexes

When LWWCM starts for the first time, it creates indexes for all objects that exist in LWWCM. When you create or update your objects, the index files will update also. If these index files become corrupted, you will be not able to see your objects even if they exist in Content Manager. To fix the indexes:

1. Stop the LWWCM application server.
2. Delete the indexes.
3. Restart the server.

The indexes will be generated again and you should be able to see your objects. The indexes are stored in: <LWWCM_HOME>\ilwwcm\system\indexes where <LWWCM_HOME> is the LWWCM home folder where you copied the /app folder during the installation.

Database alias name

If you catalog the database and you do not give it an alias name, by default, the alias name will be the same as the database name. When you define the database name of Content Manager `cm.server=[CM_Server_Name]` in the `aptrixjpe.properties` file, you need to define the alias name of the database and not the database name. When you install the connectors and the wizard asks

you for the database name, *enter the alias name* of you database. If your Content Manager system is remote (installed on a machine other than the LWWCM machine) and you cataloged the remote Content Manager Library Server database locally, enter your cataloged alias name of the database.



Using CM as a repository for LWWCM

In this chapter, we describe how to set up Content Manager as the data repository for LWWCM. In addition, we cover the initialization files that are involved to enable LWWCM to route its objects to and retrieve them from Content Manager; we show how to test the Content Manager repository connection settings, and we provide some insights into the internal Content Manager item types involved in storing LWWCM objects.

We cover the following topics in this chapter:

- ▶ Configuring LWWCM to use Content Manager as a repository
- ▶ LWWCM server routing information to access a Content Manager repository system
- ▶ How LWWCM objects are stored in Content Manager

6.1 Configuring LWWCM to use CM as a data repository

LWWCM enables users to create and manage Web sites. It is easily maintained by system administrators and easy to use by non-technical content providers.

There are many ways to store LWWCM data. By default, LWWCM stores data files in the file systems. Other options include DB2, Cloudscape, Informix, Microsoft SQL Server 2000, Oracle, and Content Manager. In this section, we focus on the configuration steps required to integrate Content Manager as the data repository with an LWWCM solution.

To set up this configuration, we need to:

1. Stop the LWWCM application server.
2. Modify the WebSphere Application Server JVM setting.
3. Modify LWWCM's `aptrixjpe.properties` file.
4. Delete existing indexes from the LWWCM server.
5. Start LWWCM and run the initial setup.

6.1.1 Prerequisites

Before configuring Content Manager as the data repository for LWWCM, you must first successfully install Content Manager and LWWCM. Make sure both systems are up and running properly. Refer to Chapter 5, "Before you begin integration: pre-integration validation" on page 63 to go through the validation checklist before you proceed with this section.

Make sure Unicode support is enabled in Content Manager Library Server database. Stop the LWWCM server.

6.1.2 Modifying the WebSphere Application Server JVM setting

There are three tasks you should perform within WebSphere Application Server:

- ▶ Add a UTF8 file coding system setting under JVM.
- ▶ Change initial and maximum Java heap size (recommendation only).
- ▶ Add or update the classpath to the JDBC driver.

Add a UTF8 file coding system setting under JVM

To do this:

1. Launch the WebSphere Application Server administration console.

2. Go to **Application Servers** -> **<LWWCM application server>** -> **Process Definition** -> **Java Virtual Machine**.

<LWWCM application server> is the application server where you install your LWWCM Web application (LWWCM.ear). By default, it is server1.

3. Create a new Custom Property. Enter `file.encoding` in the Name field, and UTF8 in the Value field.
4. Click **Apply** and **Save**.

Change the initial and maximum Java heap size (recommendation only)

It is recommended that you change the initial and maximum heap size values under the JVM setting tab. An `OutOfMemoryException` may occur if this is set too low, especially if the Web site is of significant size.

To do so:

1. Launch the WebSphere Application Server Administration console.
2. Go to **Application Servers** -> **<LWWCM application server>** -> **Process Definition** -> **Java Virtual Machine**.
3. Enter an appropriate value for the Initial Java Heap Size field. We recommend a minimum of 256 MB, or at least half of the maximum heap size value.
4. Enter an appropriate value for the Maximum Java Heap size field. We recommend 512 MB or higher.
5. Click **Apply** and **Save**.

Add or update the classpath to the JDBC driver

To be able to store LWWCM data in Content Manager, you have to install the federated connectors for Content Manager. WebSphere Application Server has to know how to communicate to the Content Manager via JDBC driver. You need to add the following to the classpath of the JDBC driver:

- ▶ The full path of the Content Manager product installation location
- ▶ A list of JAR files
- ▶ The `db2java.zip` file

To do so:

1. Launch WebSphere Application Server Administration console.
2. Go to **Application Servers** -> **<LWWCM application server>** -> **Process Definition** -> **Java Virtual Machine**.

3. Add the following to the JDBC driver's class path (in our Windows setup, we install the drivers to c:\ibm):

```
c:/ibm/Cmgmt
c:/ibm/CMBROOT/LIB/cmbview81.jar
c:/ibm/CMBROOT/LIB/cmb81.jar
c:/ibm/CMBROOT/LIB/cmbcm81.jar
c:/ibm/CMBROOT/LIB/cmbjdbc81.jar
c:/ibm/CMBROOT/LIB/cmbfed81.jar
c:/ibm/CMBROOT/LIB/cmbdb281.jar
c:/ibm/CMBROOT/LIB/cmbdj81.jar
c:/ibm/CMBROOT/LIB/cmbicm81.jar
c:/ibm/CMBROOT/LIB/cmbutil81.jar
c:/ibm/CMBROOT/LIB/cmbutiljdbc81.jar
c:/ibm/CMBROOT/LIB/cmbutilicm81.jar
c:/ibm/CMBROOT/LIB/cmbutilfed81.jar
c:/ibm/CMBROOT/LIB/xerces.jar
c:/ibm/CMBROOT/LIB/essrv.jar
c:/ibm/CMBROOT/LIB/esclisrv.jar
c:/ibm/CMBROOT/LIB/cmblog4j81.jar
c:/ibm/CMBROOT/LIB/log4j.jar
c:/ibm/CMBROOT/LIB/cmbsdk81.jar
c:/ibm/CMBROOT/LIB/cmbwas81.jar
c:/ibm/SQLLIB/java/db2java.zip
```

Note: Do not forget to substitute the provided paths (c:/ibm/Cmgmt and c:/ibm/CMBROOT) to what you have set in your environment.

6.1.3 Modifying LWWCM's *aptrixje.properties* file

You can configure LWWCM to use Content Manager to store LWWCM data by modifying *aptrixje.properties*.

Tip: *aptrixje.properties* file is located in the <LWWCM_HOME>\config directory. <LWWCM_HOME> is the Web Content Manager home folder to which you copied the /app folder during the LWWCM installation.

Locate the parameters and set their values as shown in Example 6-1.

*Example 6-1 Update *aptrixje.properties* to use CM as data repository*

```
# persistencies supported are FileSystem, JDBC and CM
manager.persistence=CM
resource.persistence=CM
...
# CM persistency settings
# CM Server information
```

```
cm.server=[CM_Server_Name]
cm.username=[CM_Username]
cm.password=[CM_Password]
cm.versionControl=false
# Unique identifier for the site
cm.jpeServerIdentifier=[CM_LWWCM_Server_ID]
```

Where:

- ▶ [CM_Server_Name] is the alias name of your Library Server. If you use a remote Content Manager system, you have to catalog the local database and use the alias name you defined. For more information on how to catalog a database see 6.2, “Accessing a remote Library Server database” on page 97.
- ▶ [CM_Username] is the content Manager user name with access to the Library Server.
- ▶ [CM_Password] is the password of this Content Manager user.
- ▶ [CM_LWWCM_Server_ID] is used to identify which LWWCM Web application’s data is stored in Content Manager. It is possible to have multiple LWWCM Web applications configured to store their data in *one* Content Manager system. To distinguish the data from these applications, each LWWCM Web application uses this CM_LWWCM_Server_ID to associate with its data. When you store LWWCM data in Content Manager, the system creates an attribute called *JPE Server ID*, which corresponds directly with CM_LWWCM_ServerID, to show that the data belongs to a specific LWWCM Web application.

Note: We recommend using the host name of the LWWCM server as the CM_LWWCM_Server_ID. If you install multiple LWWCM servers on the same physical machine, append the host name with ## where ## distinguishes the multiple LWWCM servers coming from the same host machine.

The aptrixjpe.properties file for our scenario is shown in Example 6-2.

Example 6-2 aptrixjpe.properties sample content in our scenario

```
manager.persistence=CM
resource.persistence=CM
...
cm.server=ICMNLSDb
cm.username=icmadmin
cm.password=password
cm.versionControl=false
# Unique identifier for the site
```

6.1.4 Deleting existing indexes from the LWWCM server

Once you set up Content Manager to store LWWCM data, you need to delete the existing indexes from the LWWCM server. By default, LWWCM uses a file system as the data repository. The indexes must be deleted when switching from the file system to Content Manager and vice versa so that the indexes can be recreated and consistent with the LWWCM data.

Index files are located under <WWCM_HOME>/ilwwcm/system/indexes. Delete all the files under this directory.

6.1.5 Starting LWWCM and running the initial setup

Once we delete all the existing indexes, we can restart LWWCM and run some initial setup:

1. Start the LWWCM Application server.
2. Run the AJPEGUI module. This creates the LWWCM user interface and stores all LWWCM Web site objects in the Content Manager system:

`http://[HOST]:[PORT]/[CONTEXT_ROOT]/connect/?MOD=AJPEGUI`

To verify, launch LWWCM user interface and make sure everything runs properly.

If you are migrating from an existing LWWCM system that does not use Content Manager as the data repository, you can use syndication to replicate the existing data to this new LWWCM server. For more information, refer to Chapter 13, “Data migration through syndication” on page 271.

6.1.6 Troubleshooting

When you first start up the LWWCM application server after configuring Content Manager as the LWWCM's data repository, LWWCM may not come up. In our experience, the most common reasons are an improper classpath setting in the WebSphere Application Server Administration Console for the LWWCM application server or a wrong database alias name for the remote Library Server database.

Check the error.log file in the <LWWCM_HOME>\connect\log directory for error messages containing DGL0394A or DGL0587A. If you see these error codes, you might resolve the errors as follows.

Check your Library Server database name in the `aptrixjpe.properties` file. If you use a remote Content Manager system that is installed on a machine other than your LWWCM server, you have to pay particular attention to what you specify here. Remember to always specify the name *as cataloged in DB2*. If you catalog the database with an alias name, you *must use the alias name* for the Library Server database. Also see “The `aptrixjpe.properties` file” on page 101.

Check your classpath settings specified in the WebSphere Application Server Administration Console for the LWWCM application to access the ICM connector. Check especially for the `<CMCOMMON>` path to the Content Manager and Information Integrator for Content configuration files.

Note: We recommend that you always use the Server Configuration Utility, which handles all the necessary configuration settings in the INI files. Refer to “Server Configuration Utility” on page 106 for more information.

6.2 Accessing a remote Library Server database

If the LWWCM server is installed on one machine and the Content Manager system is installed on another machine, you need to configure the LWWCM server to access the remote Content Manager system.

This may also be needed if you have a Content Manager System Administration Client on the local machine and you want to access and manage a Content Manager system on another machine.

To access a remote Content Manager system, namely, the Library Server data, you have to catalog the remote TCP/IP node and the remote Library Server database with the following DB2 command syntax:

```
CATALOG [ADMIN] TCPIP NODE node_name REMOTE host_name [SERVER service_name]
[SECURITY {SOCKS}] [REMOTE_INSTANCE instance_name] [SYSTEM system_name]
[OSTYPE os_type] [WITH "comment string"]
```

```
CATALOG DATABASE database_name [AS alias] [ON drive | AT NODE node_name]
[AUTHENTICATION {SERVER | CLIENT | DCS | DCE SERVER PRINCIPAL principalname
|
KERBEROS TARGET PRINCIPAL principalname | SERVER_ENCRYPT | DCS_ENCRYPT}]
[WITH "comment_string"]
```

Note: `SERVER service_name` is mandatory for `CATALOG TCPIP NODE` and not allowed for `CATALOG ADMIN TCPIP NODE`.

In our scenario, we use the following commands to access a remote Library Server database on a Windows machine demo2 and on an AIX machine named zaire:

- ▶ `catalog tcpip node <node> remote <host name> server <port>`
`catalog TCP/IP node DEMO2 remote demo2.svl.ibm.com server 50000`
`catalog TCP/IP node zaireaix remote zaire.svl.ibm.com server 60002`
- ▶ `catalog database <remote db name> as <alias> at node <node name>`
`catalog database ICMNLSDB as DEMO2LDB at node DEMO2`
`catalog database ICMNLSDB as ZDB at node zaireaix`

For the node name, choose the name you want, but restrict it to eight characters. To find the right port number of the remote Library Server:

- ▶ On AIX:
 - a. Go to the remote system's /etc subdirectory, and enter: `cat services`.
 - b. Get the DB2 instances by entering `db2 get instance`.
 - c. Find the port number for the entry `db2c_<instance name>` in the services file. This is the listening port for the database instance for this remote Library Server database.
- ▶ On Windows:
 - a. Go to the subdirectory `C:\WINNT\system32\drivers\etc` and enter type `services`.
 - b. In the DB2 command window, enter `db2 get instance`.
 - c. Find the port number for the entry `db2c_<instance name>` in the services file. This is the listening port for the database instance for this remote Library Server database.

Also check the DB2COMM configuration parameters with **db2set** in a DB2 command window. If you use TCP/IP communication, check to see if it is configured:

```
DB2COMM=TCPIP, NETBIOS
```

Test the connection by entering a **db2 connect** command from the DB2 command window. In our example, we try a connection against the remote AIX database ICMNLSDB, alias ZDB:

```
db2 connect to ZDB user ICMADMIN using password
```

Note: If you get an SQL error with SQLSTATE=08001, refer to 5.4, “Problem determination and troubleshooting” on page 78 and specifically 5.4.3, “DB2 and NSE” on page 80.

If you are not familiar with DB2 commands, use the DB2 Control Center to access a remote DB2 system and to catalog the remote Library Server database:

1. Select **Start -> Program -> IBM DB2 -> General Administration Tools -> Control Center**.
2. Right-click **Systems** and select **Add** from the context menu (see Figure 6-1).
3. Enter the appropriate values for the new system.
4. Right-click the **Databases** folder and select **Add** from the context menu.
5. Enter the appropriate values for the remote database.

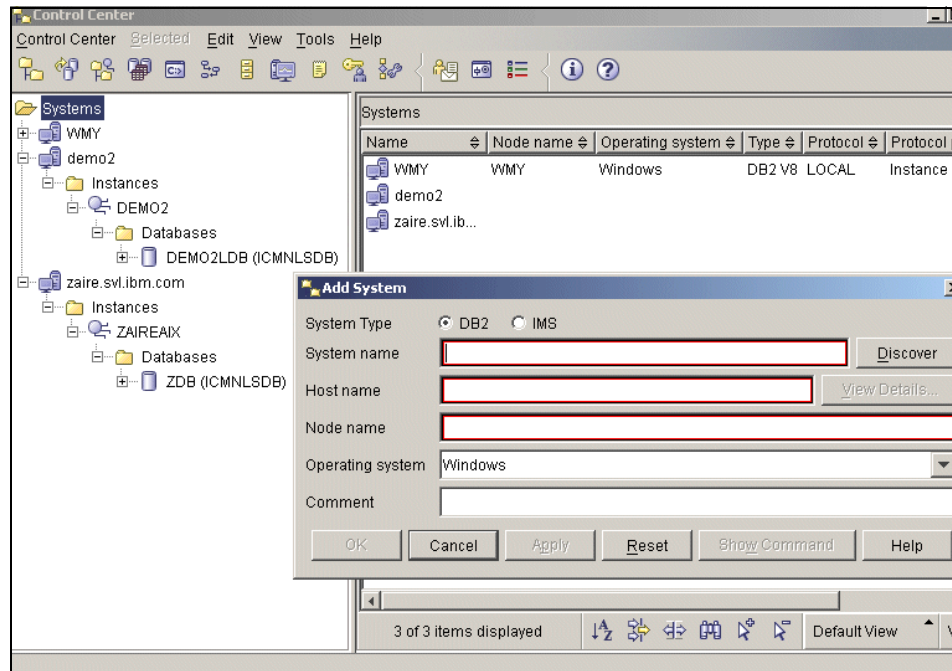


Figure 6-1 DB2 Control Center

6.3 LWWCM server routing information to access CM

In this section, we describe how LWWCM uses server routing configuration files and connectors to access a Content Manager system. It is important to understand the internal details involved. If you run into setup troubles or want to perform a non-typical configuration, this in-depth knowledge will help you.

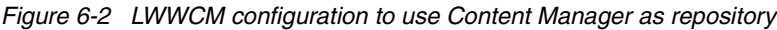
The LWWCM server that uses Content Manager as the data repository needs the Content Manager (ICM) connector and the Information Integrator for Content

(FED) connector to be locally installed on the LWWCM server. The connectors can be installed as part of the Information Integrator for Content Development Workstation installation. The ICM connector is necessary when you run the AJPEGUI module to create the LWWCM user interface. In addition, during LWWCM Web solution development, the FED connector is necessary to store and to retrieve the LWWCM objects stored in a local or remote Content Manager system. The FED connector and the ICM connector are needed if you use the LWWCM federated content component. The LWWCM federated content component allows LWWCM users to perform searches for federated content, attributes and content documents and to integrate the results into LWWCM content pages (see Chapter 7, “Publishing Content Manager content” on page 115).

To ensure that the LWWCM and Content Manager integration solution works, you must ensure that you specify the right routing configuration information. Routing information means network information for LWWCM server(s) to access a Content Manager repository or Information Integrator for Content database (which is used with LWWCM's federated content component). The setup and routing information are specified in the following three files:

- ▶ aptrixjpe.properties
- ▶ cmbicmsrvs.ini
- ▶ cmbds.ini

These files are necessary so the LWWCM server knows from where to retrieve or where to store the LWWCM objects. Multiple LWWCM servers may share one Content Manager system as a repository. Figure 6-2 on page 101 presents an overview of how these files are involved in an LWWCM and Content Manager integration solution.



In this figure, the installation and setup is done so that the Information Integrator for Content administration database is shared with the Content Manager Library Server database. This allows you to manage a single set of user IDs for both Content Manager and Information Integrator for Content. A setup such as this might imply a risk because, if you recreate this database, you would lose LWWCM data and Information Integrator for Content data (such as federated search templates). Of course, information loss can be prevented by implementing a back-up or fail-over environment, which is always essential for any production setup.

6.3.1 The aptrixjpe.properties file

The *aptrixje.properties* file is used to define the LWWCM data storage type. The location of this file is specified during the LWWCM server installation. It contains parameters that can be edited to change the behavior of an LWWCM environment. In this redbook, we focus on using Content Manager as data storage. Other storage types supported are File System or JDBC storage systems such as Cloudscape, Informix, Microsoft SQL Server 2000, and Oracle.

Whenever you change your existing LWWCM storage, for example the file system, to a new storage like Content Manager, follow the steps described in Chapter 13, “Data migration through syndication” on page 271.

If you specified Content Manager as the repository type in the `aptrixjpe.properties` file and have not launched the LWWCM building module program (AJPEGUI) and its user interface, you can just start the WebSphere Application Server and enter the following URL in the Web browser:

```
http://<hostname>:<port>/<context_root>/connect/?MOD=AJPEGUI
```

This runs the LWWCM's AJPEGUI building module program, creates the LWWCM user interface, and stores all LWWCM item types in the Content Manager system. Afterwards, the user can run the LWWCM user interface.

You can optionally set many parameters in the `aptrixjpe.properties` file. We describe the parameters that are related to sharing one Content Manager repository across multiple LWWCM server instances.

There are two possibilities within the `aptrixjpe.properties` file to achieve this:

- ▶ Set each LWWCM server to use a unique server identifier:

```
cm.jpeServerIdentifier=<unique_server_ID>
```

- ▶ Set each LWWCM server to use different item types that will be stored in Content Manager. The default is as follows:

```
cm.itemType=AJPEData  
cm.resources.itemType=AJPEResources
```

In the first approach, you specify a *unique `cm.jpeServerIdentifier`* parameter for the current LWWCM server instance in the `aptrixjpe.properties` properties file.

In the second approach, which should only be used by experienced Content Manager administrators, you specify the name of the Content Manager item types used for LWWCM data and resources. This is done by specifying the parameters *`cm.itemType`* and *`cm.resources.itemType`*.

With the second approach, it is the responsibility of the Content Manager system administrator to specify the item types to store the LWWCM data and resources objects. The default LWWCM item type names in Content Manager are `AJPEDate` and `AJPEResources`.

We recommend using the first approach (that is, a unique server identifier using default item types) to distinguish between different LWWCM servers' data within one Content Manager repository.

The second approach might be useful if you want to use different Content Managers' Resource Managers to store data from different LWWCM servers. Remember that the LWWCM files and image objects are stored as `AJPEData`, `AJPEResources` items and the physical binary objects are stored in Content Manager's Resource Manager.

For more information on how to set up and configure an integrated LWWCM and Content Manager solution, refer to Chapter 8, “System architecture considerations” on page 167.

For more information on parameter settings in the `aptrixjpe.properties` file, refer to LWWCM’s online Information Center.

6.3.2 Files `cmbicmsrvs.ini` and `cmbds.ini`

The `cmbicmsrvs.ini` and `cmbds.ini` files are installed within the Information Integrator for Content installation. Shown in Figure 6-3 is the component selection window during the product installation. Make sure the Content Manager V8 (ICM) connector and the federated (FED) connector are selected.

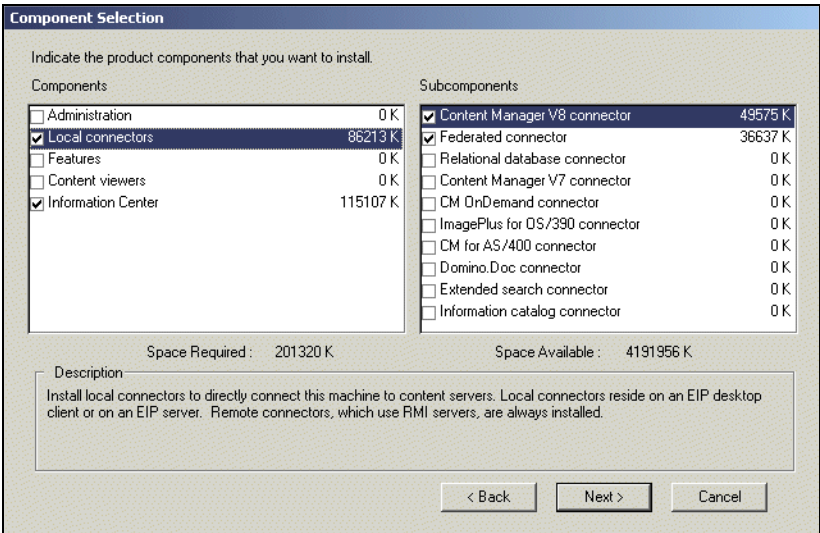


Figure 6-3 II for Content ICM and FED connector installation

The INI files are located in the `<CMCOMMON>` directory. On Windows, the default is `c:\cmgmt` directory. On AIX, the default is `/usr/lpp/cmb/cmgmt`. The `cmgmt` subdirectory contains all common Content Manager configuration files. “Common” refers to the information used by Content Manager and the Information Integrator for Content.

File `cmbicmsrvs.ini`

The `cmbicmsrvs.ini` file contains the Library Server database information. It has one entry for each Library Server that we want to access from an LWWCM server installation. Whenever you catalog a new Library Server database (see 6.2,

“Accessing a remote Library Server database” on page 97), you also need to add the Library Server database information to this file.

If the cmbicmsrvs.ini is misconfigured or the Library Server is not reachable, then the LWWCM and Content Manager integrated solution would not work with an LWWCM user interface. When you start the LWWCM user interface, you will receive the following error in your browser window:

ERROR 503 [LWWCM Framework] reported an error

The information in the INI file, if available, is ready by the WebSphere Application Server server1 LWWCM application when the WebSphere Application Server is started.

In our setup, we also use the Content Manager System Administration Client and the Windows Client to define new item types and to import item objects. We have more than just the LWWCM-related Library Server entry. The reason is that we test different system setups with local and remote Content Manager systems. In our scenario, our cmbicmsrvs.ini file contents on a Windows machine are shown in Example 6-3.

Example 6-3 cmbicmsrvs.ini sample content in our scenario

```
CMSEVER=ICMNLSDB
ICMSERVERREPTYPE=DB2
ICMSHEMA=ICMADMIN
ICMSSO=FALSE
ICMDBAUTH=SERVER
ICMREMOTE=FALSE
ICMHOSTNAME=
ICMPORT=
ICMREMOTEDB=
ICMNODENAME=
ICMOSTYPE=
```

```
ICMSERVER=DEM02DB
ICMSERVERREPTYPE=DB2
ICMSHEMA=ICMADMIN
ICMSSO=FALSE
ICMDBAUTH=SERVER
ICMREMOTE=TRUE
ICMHOSTNAME=DEM02
ICMPORT=
ICMREMOTEDB=ICMNLSDB
ICMNODENAME=DEM02
ICMOSTYPE=WIN
```

```
ICMSERVER=ZDB
ICMSERVERREPTYPE=DB2
```

```
ICMSchema=ICMADMIN
ICMSSO=FALSE
ICMDBAUTH=SERVER
ICMREMOTE=TRUE
ICMHOSTNAME=zaire
ICMPort=50000
ICMREMOVEDB=ICMNLSDb
ICMNODENAME=ZAIREAIX
ICMOSTYPE=AIX
```

The Library Server database entry, ICMNLSDb, in the first block is the local (ICMREMOTE = FALSE) LWWCM server Content Manager datastore. This entry is also referenced in the `aptrixjpe.properties` file containing the LWWCM configuration parameters:

```
cm.server=[CM_SERVER_NAME]
```

In our scenario setup, we use the following in the `aptrixjpe.properties` file:

```
cm.server=ICMNLSDb
```

The `ICMSERVER=DEMO2LDB` (alias name as specified with the **db2 catalog database** command) entry is used to access a remote Library Server database on a remote Windows machine named *demo2* and the `ICMSERVER=ZDB` entry to access a remote Library Server database on a AIX machine named *zaire*.

To specify the right values in the `cmbicmsrvs.ini` file, you can also use the Server Configuration Utility, as detailed in “Server Configuration Utility” on page 106.

File cmbds.ini

The *cmbds.ini* file contains the federated server database list for Information Integrator for Content. It is used when you want to use federated search templates, which you have previously created with the Information Integrator for Content Administration Client, to create federated content components in LWWCM. Refer to 7.3, “Federated content component” on page 123 for information on federated content component.

This federated server database list is read by the WebSphere Application Server server1’s LWWCM application when the WebSphere Application Server is started. The database connection can be defined by cataloging the database using the DB2 Configuration Assistant or the DB2 command line. You can also use the Server Configuration Utility. Refer to “Server Configuration Utility” on page 106.

If the information is not available when the WebSphere Application Server server1’s LWWCM application is started, you will receive an error when you create a new federated content component in LWWCM (**Design & Development**

-> **Component Library -> New -> Select Federated Content Component -> OK -> Content Reference -> Select Reference**):

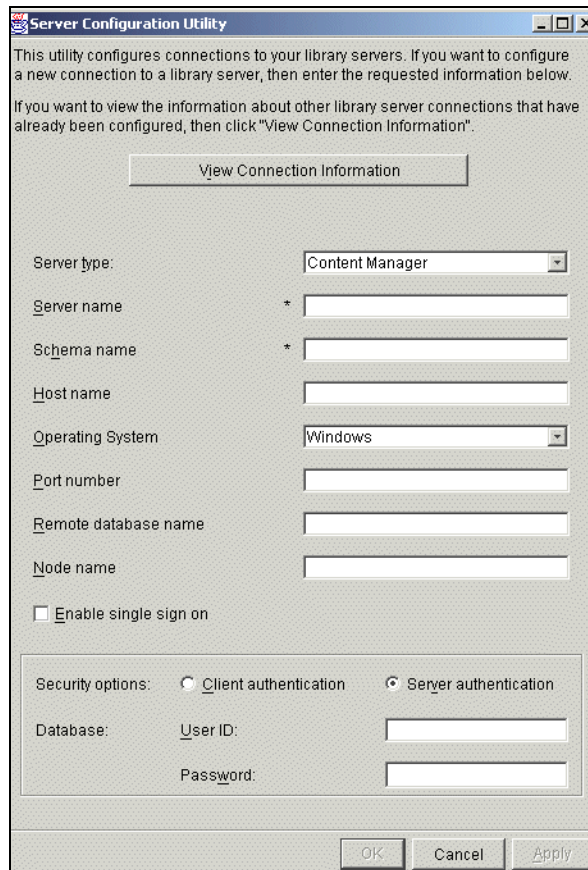
Couldn't connect to remote host.

Or, if you try to preview an existing federated content component object, you will receive the following error in the browser window:

The page cannot be found.

Server Configuration Utility

To specify the right information or to correct the information in the cmbicmsrvs.ini and cmbds.ini files regarding access to a Content Manager Library Server database or an Information Integrator for Content database, you can also use the Server Configuration Utility, as shown in Figure 6-4.



The screenshot shows a Windows-style dialog box titled "Server Configuration Utility". It contains instructional text at the top, a "View Connection Information" button, and several input fields for configuring a library server connection. The fields include "Server type" (a dropdown menu set to "Content Manager"), "Server name", "Schema name", "Host name", "Operating System" (a dropdown menu set to "Windows"), "Port number", "Remote database name", and "Node name". There is also a checkbox for "Enable single sign on". At the bottom, there are "Security options" with radio buttons for "Client authentication" and "Server authentication" (which is selected). Below these are fields for "Database: User ID:" and "Password:". At the very bottom are "OK", "Cancel", and "Apply" buttons.

Figure 6-4 Server configuration utility

Start the Windows utility by selecting **Start -> Programs -> IBM Content Manager for Multiplatforms V8.2 -> Server Configuration** or **Start -> Programs -> EIP for Multiplatforms -> Server Configuration**. This tool is installed with the Content Manager System Administration or Information Integrator for Content System Administration installation. The utility copies all the necessary linformation, such as the database schema name, alias name, operating system to the cmbds.ini and/or cmbicmsrvs.ini file. If you choose **View Connection Information**, it will bring up all (using the **db catalog** command) the configured databases. On the Server type drop-down list, you can choose whether you want to specify a new entry in the cmbds.ini (choose **Enterprise Information Portal**, which is Information Integrator for Content) or a new entry block in the cmbicmsrvs.ini file (select Content Manager).

6.3.3 Testing the ICM and FED connectors

To test the Content Manager and Information Integrator for Content server settings on Windows or AIX, we run a sample. The prerequisite is that the Content Manager online Information Center must be installed. This can be selected during a Content Manager or Information Integrator for Content System Administration installation. The installation includes Content Manager and Information Integrator for Content sample programs that test different Content Manager and Information Integrator for Content system functionalities.

We use the TConnect.java sample program to connect and disconnects from Content Manager and Information Integrator for Content databases as configured in the cmbds.ini and cmbicmsrvs.ini file:

1. Check and, if necessary, change the settings in a file named cmbenv81.bat for CMBROOT, DB2HOME, JAVAHOME, WAS_HOME variable settings to the installation directories in your system. This tool can be found on Windows in the <CMBROOT> subdirectory and on AIX named cmbenv81.sh in the /usr/lpp/cmb/bin subdirectory.
2. Run cmbenv81.bat (Windows) or cmbenv81.sh (AIX). Go to the <CMBROOT>/samples/java/beans (on AIX) and to the <CMBROOT>\infoctr\info\apiinfo\dk\samples\java\bean (on Windows) subdirectory.
3. Compile the TConnect java file with **javac TConnect.java**.
4. Run the TConnect.class file.

The syntax of the TConnect class is:

```
java TConnect [-displayname] <dstype> <server> <userid> <password>
where:
```

[-displayname] is a flag to use display names. Default is nondisplay names.

<dstype> is the type of server (Fed, ICM, DL, OD, DB2)

<server> is the name of the server or database. This can be followed by a connect string in parenthesis.
<userid> is the userid
<password> is the password

Examples:

```
java TConnect -displayname ICM icmnlbdb icmadmin password
java TConnect DB2 sample db2admin password
java TConnect Fed cmbdb cmbadmin password -displayname
java TConnect OD 9.112.20.3(ODPORT=2190) userid password
```

Example 6-4 shows the testing and the result of running TConnect to test in our AIX system setup:

Example 6-4 TConnect run result in our AIX scenario

```
zaire|icmadmin>$ java TConnect ICM icmnlbdb icmadmin <password>
Connecting to server
Disconnecting from server
zaire|icmadmin>$
```

6.4 How LWWCM objects are stored in CM

After a new LWWCM server installation, when you call LWWCM's AJPEGUI module by entering the following URL from a browser, all the LWWCM objects are created in Content Manager:

`http://[HOST]:[PORT]/[CONTEXT_ROOT]/connect/?MOD=AJPEGUI`

Two new Document item type classification objects are created in the Content Manager Library Server, *AJPEData* and *AJPEResources*, with the attributes as shown in Figure 6-5. The item type properties of the two item types can be viewed within the Content Manager System Administration Client window.

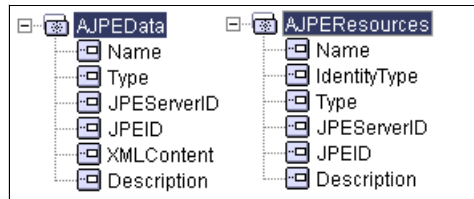


Figure 6-5 AJPEData and AJPEResources item type attributes

6.4.1 Viewing AJPEData and AJPEResources item types

You can view the values of AJPEData and AJPEResource item types by using the Content Manager System Administration Client:

1. Select **Start -> Programs -> IBM Content Manager for Multiplatforms V8.2 -> System Administration Client**.
2. Log on to the Content Manager Library Server database as specified in the aptrixjpe.properties file. In our example, it is cm.server=icmnlsdb.
3. Open the Library Server database folder and then select **Data Modeling -> Item Type**.
4. Right-click **AJPEResources** and select **Properties** from the context menu.
5. On the item type properties screen, select a different panel, such as Document Management (see Figure 6-6) to see how the item types are defined.

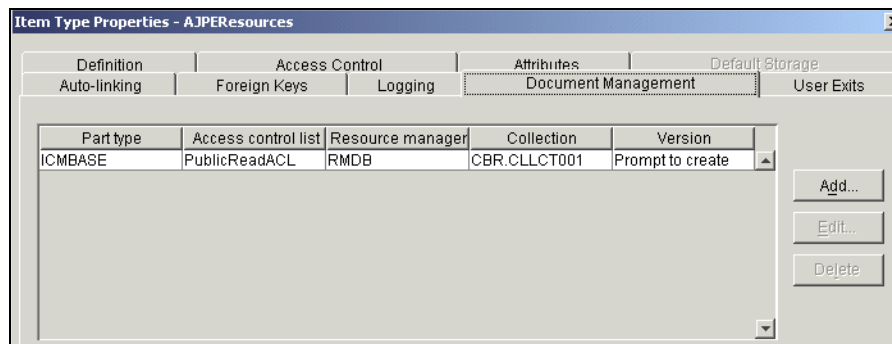


Figure 6-6 AJPEResource document management definition

Note: All Content Manager system objects are stored in DB2 tables. In our example, we use ICMNLSDB as the Library Server database and RMDB as the Resource Manager database. All definitions you see in the Content Manager System Administration Clients window are SQL queries to the ICMNLSDB and RMDB tables. RMDB is queried when you navigate down the Resource Manager tree. The client does an secure socket layer (SSL) connect to RMDB to get the System Managed Storage (SMS) definitions.

6.4.2 Storing LWWCM objects in AJPEData/AJPEResources objects

Each LWWCM object information is stored as a Content Manager AJPEData item. LWWCM Component Library file resource or image component objects are each stored in an Content Manager AJPEData item *and* an AJPEResources

Content Manager item. In addition, the binary files or image objects are stored in the Content Manager Resource Manager. See Figure 6-7 for details.

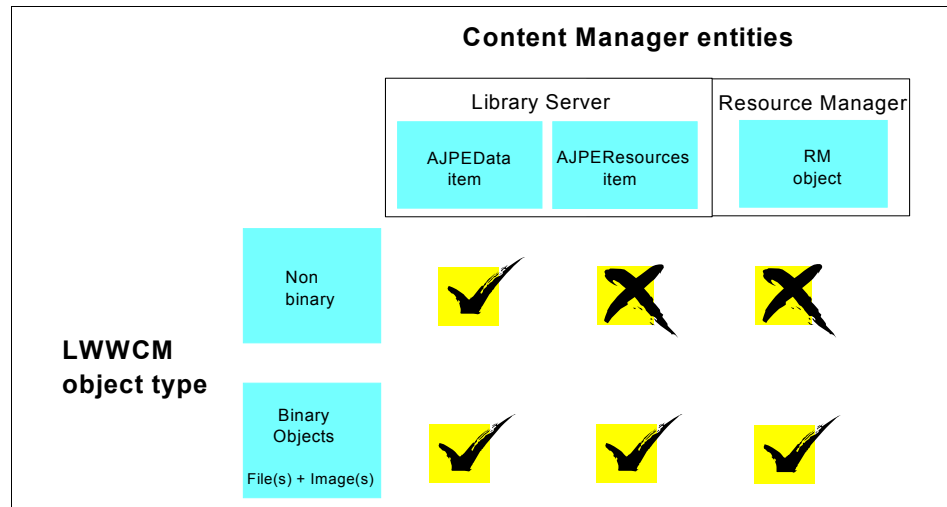


Figure 6-7 How LWWCM objects are stored in Content Manager

LWWCM objects which are stored as Content Manager AJPEData entity are:

- ▶ Library content, template, page design, file and image objects, search rules, site, site area, site, site area, taxonomy, category, users and groups, workflow, workflow stage, workflow action, syndication, subscriber and objects.
- ▶ Component library objects that include text, menu, navigator, rich text, HTML, reference, taxonomy, user name, federated, JSP components.

LWWCM objects which have a Content Manager AJPEData, AJPEResources and Resource Manager entries are the LWWCM Component Library objects:

- ▶ File resource component
- ▶ Image component

6.4.3 Viewing AJPEData and AJPEResources items

With the Content Manager Windows Client or eClient, you can do a basic search for AJPEData item types. In Figure 6-8 on page 111, you can see the results of a basic search.















AJPEData					
	Name	Type	JPE Server ID	JPE ID	Description
	All Items	itemGatherers	demo2	fa4a25691...	
	All Live Items	itemGatherers	demo2	5d6709ca9...	
	Administrator	security	demo2	4ffefdb691...	
	Administrators	security	demo2	4022a6509...	
	UI Access Group	security	demo2	5967260d9...	
	[ALL USERS]	security	demo2	620d287a9...	A group containing all users
	Version Creators	security	demo2	260812259...	
	Restorers	security	demo2	1b54c0f19...	
	Anonymous	security	demo2	b8c210c19...	
	Publish	actions	demo2	564fb7c49...	Publishes a document
	Expire	actions	demo2	a23d29379...	Expires a document
	Publish and Version	actions	demo2	531d50a59...	Publish and save a version of a document
	Current Content Categories	profileMappers	demo2	a6acda419...	Maps to the current content category profile
	Current User Categories	profileMappers	demo2	6f5e851d9...	Maps to the current user category profile

Figure 6-8 Content Manager AJPEData items after LWWCM-Content Manager setup

If you create a new file or image LWWCM component, you also receive a search result for the AJPEResources Content Manager item type. In our setup, the search result for Content Manager AJPEResources items from a “DEMO2” machine Content Manager setup is shown in Figure 6-9.











AJPEResources						
	Name	Identity Type	Type	JPE Server ID	JPE ID	Description
	investment1.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c02d406f63a97	
	investment.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c02e406f6402f	
	investment2.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c042406fd0fc9	
	investment3.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c043406fd116d	
	Mother&Doughter.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c05f4070b773c	
	familydraw.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c0604070b7962	
	Family.jpg	com.aptrix.pluto.resource.ImageResource	Resource Byte Content	demo2	468c07440a5827e6	
	BR028_CS_Overview.PDF	com.aptrix.pluto.resource.FileResource	Resource Byte Content	demo2	eb000007eb8b87c	
	SteelBlue.css	com.aptrix.pluto.resource.FileResource	Resource Byte Content	demo2	eb000107eb8c013	
	BR035_Jump_IT.PDF	com.aptrix.pluto.resource.FileResource	Resource Byte Content	demo2	eb000207eb8cdec	

Figure 6-9 Windows client AJPEResources search result

Important: Never change any LWWCM items from the Content Manager side; otherwise, LWWCM objects or your LWWCM Web site could become inconsistent and the LWWCM user interface with the existing LWWCM objects could become unusable.

AJPEData

Items of the AJPEData item type have the following attributes:

- Name: Contains the title of the LWWCM objects as is viewed under the LWWCM Federated Content Component -> Image Component under Title field (see Figure 6-10 on page 112).

- ▶ **Type:** Can be of type cmpnts, resources, security, styles, profileMappers, templates, site area, actions, itemGatherers, sites, contents, categories, taxonomy, workflows, and stages.
- ▶ **JPEDServerID:** Is a unique identifier for every LWWCM server and is specified in the aptrixjpe.properties file as cm.jpeServerIdentifier=demo2 in our example.
- ▶ **JPEID:** Every LWWCM object created in any LWWCM server gets a unique ID. This ID is the link ID to the corresponding AJPEResource item.
- ▶ **Description:** As entered during the creation of an LWWCM object in the Description entry field.

Title	Status	Date Modified	Authors
Components By Type			
Federated Content Component			
CLAIMPhotos	o o o	Jun 14, 2004 6:08:10 PM PDT	icmadmin
CM cmpnt 2	o o o	Jun 1, 2004 3:13:46 PM PDT	icmadmin
CM comp test	o o o	Jun 16, 2004 9:20:03 AM PDT	icmadmin
File Resource Component			
Image Component			
Walters contract image	o o o	Jun 18, 2004 1:47:33 PM PDT	icmadmin

Figure 6-10 LWWCM component library. image component

AJPEResources

This is the item type where the LWWCM files or image objects can be viewed if selected and double-clicked from the Content Manager Windows Client or eClient. Items of AJPEResources item type have the following attributes:

- ▶ **Name:** Name of the LWWCM image or file resource objects as entered in the File entry field when the federated component object is created.
- ▶ **Identity Type:** In LWWCM V2, this is com.aptrix.pluto.resource.ImageResource or com.aptrix.pluto.resource.FileResource identity type.
- ▶ **Type:** In LWWCM V2, only the Resource Byte Content type exists.
- ▶ **JPEDServerID:** Is a unique identifier for every LWWCM server and is specified in the aptrixjpe.properties file as cm.jpeServerIdentifier=demo2 in our example.
- ▶ **JPEID:** Every LWWCM object from a server has a unique ID. This ID is the link ID to the corresponding AJPEData item.
- ▶ **Description:** No entry can be specified in the LWWCM user interface to be shown in this field. It is always blank.

6.4.4 Viewing AJPEData XMLContent data

The attribute, XMLContent, in AJPEData is a serialized LWWCM Java object. It represents the metadata for the LWWCM objects. All LWWCM objects have XMLContent attribute values in Content Manager's AJPEData entry.

Search in the ICMSTCOMPVIEWDEFS Library Server database table for AJPEData001. In a DB2 command line window, this can be done by entering the following DB2 commands:

```
db2 select COMPONENTTYPEID,COMPONENTVIEWNAME from  
ICMADMIN.ICMSTCOMPVIEWDEFS where COMPONENTVIEWNAME='AJPEData001'
```

In our example, the SQL query result is:

```
COMPONENTTYPEID COMPONENTVIEWNAME  
-----  
1018 AJPEData001
```

1 record(s) selected.

After searching for a table named "ICMUT*..." containing the COMPONENTTYPEID, we find a table named ICMUT01018001. To see how the AJPEData XMLContent attribute is stored in the ICMUT01018001 table, run a db2 command such as in our example:

```
db2 select * from ICMADMIN.ICMUT01018001 > XMLContent.txt
```




Publishing Content Manager content

In this chapter, we describe various ways of publishing Content Manager content to LWWCM. We address how LWWCM retrieves and publishes this information on a Web site.

The topics we cover include:

- ▶ Prerequisites
- ▶ Federated content component
- ▶ Connect tags
- ▶ JSP component

7.1 Introduction

LWWCM can retrieve data from a variety of external sources and publish them through Web sites. In this type of integration scenario, LWWCM is used as the front end for the administration and maintenance of Web sites. The data is created from different applications (such as Content Manager) and stored in external repositories. LWWCM performs data retrieval from the external sources. The retrieved data is for display on the Web sites only; it cannot and should not be modified from LWWCM. It is somewhat like a snapshot of the data source. The only way to modify this data is from the original source where the data is created.

Why retrieve external data?

In order to automatically maintain the latest and most up-to-date information from different sources and providers, it is logical to keep the data where it is usually created and retrieve it for display to the Web sites as needed.

How to retrieve data from external sources

We can retrieve data using three different approaches from LWWCM. These approaches are based on the use of the following components:

- ▶ Federated content components: used to incorporate one single external record into your Web page. A single record can be a video, a document content or all or a subset of one item object metadata set (attributes). The LWWCM federated content component facilitates the search for the references of that content through the use of the search templates created in Information Integrator for Content Administration Client.
- ▶ Connect tags: used to create lists of external resources based on a back-end specific query string. For example, they can be used to show a list of URL references on the Web site.
- ▶ JSP components: the most powerful and flexible way to integrate the external Content Manager resources by leveraging, for example, the Content Manager Java APIs.

We discuss each of these components in the following sections of this chapter. There are two ways of using them:

- ▶ Use the components as *shared resources* stored in the Component Library.
This means that we create the components and store them as shared repositories. These components can then be used multiple times.
- ▶ Use the components to include the results within the content.
This way, the components cannot be shared and used by others.

We can also use the components to display retrieved data within specific LWWCM content, combined in the same Web page.

7.2 Prerequisites

In order to use the federated content components, the JSP components, or the connect tags, you must verify the following prerequisites for both LWWCM and Content Manager servers.

7.2.1 LWWCM server

On the LWWCM server, it is mandatory to install the Content Manager Information Integrator for Content JAR files. This can be done by installing the Information Integrator for Content Development Workstation (Figure 7-1) and the needed local connectors (Figure 7-2 on page 118).

Please also refer to Chapter 5, “Before you begin integration: pre-integration validation” on page 63 for more information.

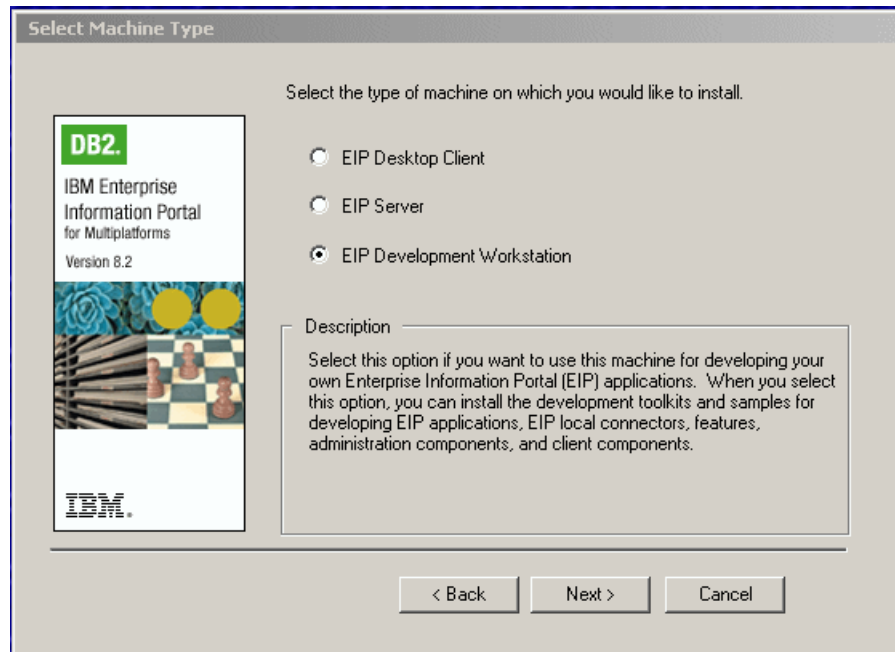


Figure 7-1 Installation II for Content: development workstation

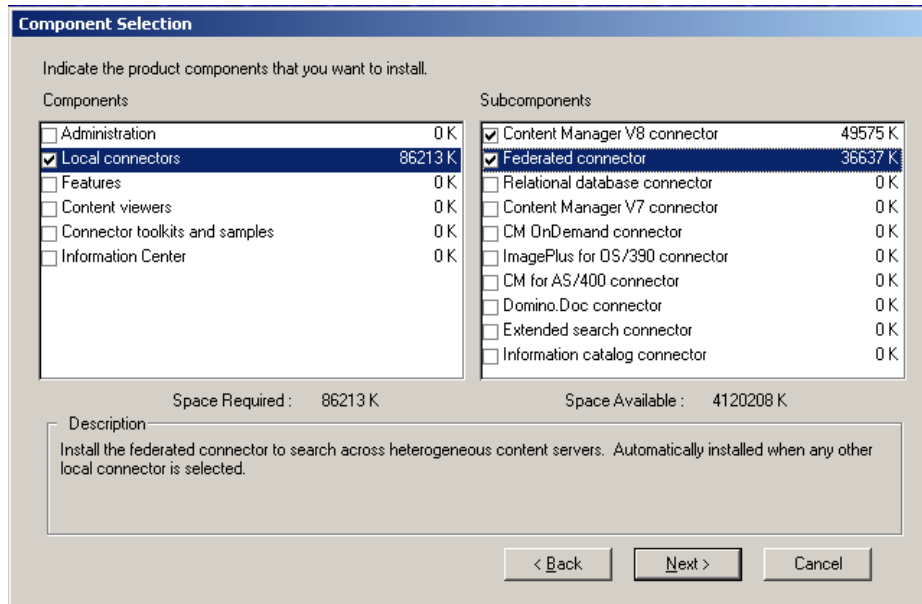


Figure 7-2 Installation Information Integrator for Content: local connectors

The classpath of the application server on which you intend to put the JSP files has to include the JAR files necessary to access Content Manager. These are the same JAR files that are needed when using Content Manager as the repository for LWWCM, since the API access to the Content Manager data is realized using the Information Integrator for Content connector technology.

7.2.2 Content Manager server

Depending on which connector technology you want to use, it might also be necessary to install Information Integrator for Content on the Content Manager server. For example, if you use the federated content component, you must create search templates on the Content Manager server. This means that you must have the Information Integrator for Content installed on the server.

7.2.3 Content Manager data

In order to be able to retrieve data from Content Manager to LWWCM, you should have some data in your Content Manager server. For our case study, we create the Content Manager data model and import some sample data to the Content Manager system.

The steps are as follows:

1. Creating new attributes in Content Manager
2. Creating a new item type in Content Manager
3. Creating new content in Content Manager

Creating new attributes in Content Manager

The attributes we create here work with our insurance case study. Table 7-1 shows a list of attributes that we need to create.

Table 7-1 Attributes in Content Manager

Attribute name	Attribute type	Character type	Length
IT_Name	Character	Other	50
IT_Number	Short integer		
IT_Description	Character	Other	250
IT_TargetGroup	Character	Other	50

To create each of them, you should follow these steps:

1. Launch System Administration Client by selecting **Start -> Programs -> IBM Content Manager for Multiplatforms V8.2 -> System Administration Client**.
2. Log in with a proper username and password.
3. From the left-hand navigator, expand **Data Modeling**.
4. Right-click **Attributes** and select **New** from the context menu.
5. Enter IT_Name for the Name field.
6. Select the proper attribute type and the character type.
7. Enter 50 for the character length (see the setup in Figure 7-3 on page 120).
8. Click **OK** to save the attribute definition.
9. Repeat the above steps for the remaining attributes.

New Attribute

Name: * ITName

Display name: * ITName Translate...

Attribute type

- ☒ Character
- ☐ Variable character
- ☐ Short integer
- ☐ Long integer
- ☐ Decimal
- ☐ Double
- ☐ Date
- ☐ Time
- ☐ Time stamp
- ☐ BLOB
- ☐ CLOB

Character type

- ☐ Alphabetic (1)
- ☐ Numeric (2)
- ☐ Alphanumeric (3)
- ☐ Extended alphanumeric (4)
- ☒ Other (5)

Character length

Length: 50

OK Cancel Apply Help

Figure 7-3 New attribute

Creating a new item type in Content Manager

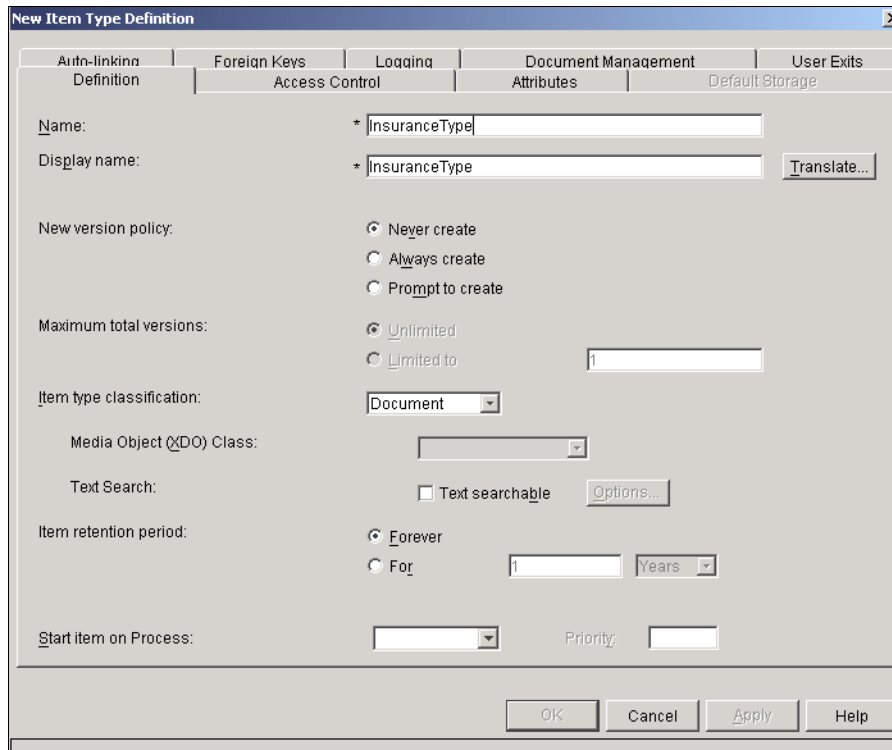
Following the case study from Chapter 4, “Case study” on page 41, we create a new item type within Content Manager for the insurance company.

To create a new item type, follow these steps:

1. In the System Administration Client, expand **Data Modeling**.
2. Right-click **Item Type** and select **New** from the context menu.
3. Set up information in the Definition tab:

Enter Insurance Type for the Name and Display name fields (see Figure 7-4 on page 121).
4. Set up information in the Attribute tab:
 - a. Select the **Attributes** tab.
 - b. Add the attributes created in the previous section by selecting them from the left panel and clicking **Add**. See Figure 7-5 on page 121 for the resulting selected attributes.

c. Click **Apply**.

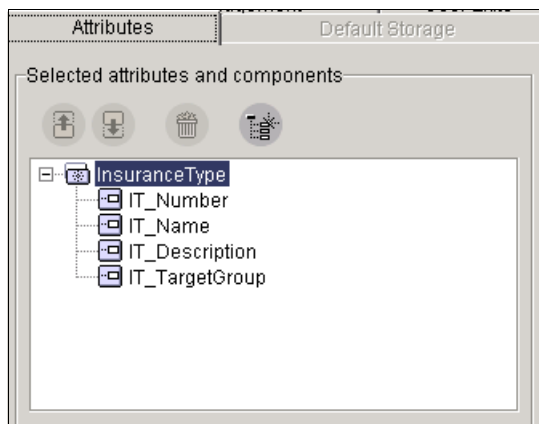


The 'New Item Type Definition' dialog box is shown with the 'Definition' tab selected. It contains the following fields and options:

- Name:** * InsuranceType
- Display name:** * InsuranceType (with a 'Translate...' button)
- New version policy:** ☒ Never create, ☐ Always create, ☐ Prompt to create
- Maximum total versions:** ☒ Unlimited, ☐ Limited to (with a text field containing '1')
- Item type classification:** Document (dropdown)
- Media Object (XDO) Class:** (empty dropdown)
- Text Search:** ☐ Text searchable (with an 'Options...' button)
- Item retention period:** ☒ Forever, ☐ For (with a text field containing '1' and a 'Years' dropdown)
- Start item on Process:** (empty dropdown), **Priority:** (empty text field)

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 7-4 New Item Type window



The 'Item Type attributes' dialog box is shown with the 'Attributes' tab selected. It displays a tree view of selected attributes and components for the 'InsuranceType' item type:

- InsuranceType
 - IT_Number
 - IT_Name
 - IT_Description
 - IT_TargetGroup

Buttons at the top: Add, Remove, Delete, Copy.

Figure 7-5 Item Type attributes

5. Set up Document Management information:

- a. Select the **Document Management** tab.
 - b. Click **Add**.
 - c. Select **ICMBASE** from the Part type drop-down list.
 - d. If you want to use versioning, change the New version policy to **Always create** or to **Prompt to create**.
 - e. Do not change the other default values (see Figure 7-6). Click **OK** and then **Apply**.
6. Click **OK** to save this new item type.

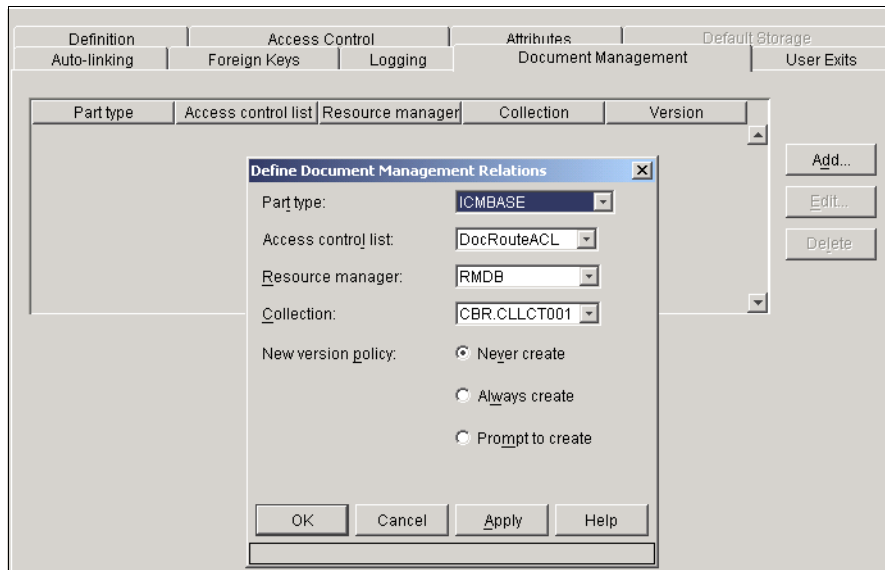


Figure 7-6 Item Type Document Management window

Creating content in Content Manager

Once we have created attributes and an item type for our insurance case study, we need to create (import) content into the Content Manager system. This content will be used in the federated searches in our case study. It could include the different insurance plans documented using Microsoft Word or other word processors.

To create content in Content Manager, follow these steps:

1. Start Content Manager Client for Windows by selecting **Start -> Programs -> IBM Content Manager V8 -> Client for Windows**.
2. Log in with a proper username and password.
3. Click **Import**.

4. Click **Add Files to Import**.
5. Select a file (for instance, a text file with insurance information) and click **OK**.
6. Select **Insurance Type** in the Item Type field, and enter values for the attributes fields.
7. Click **Import**.
8. Repeat the above steps to import more documents into the Content Manager system.

7.3 Federated content component

There are three ways of retrieving data from Content Manager and publishing the data to LWWCM Web sites. One way is to use the federated content component. In this section, we discuss what it is, why we use it, how to set it up with sample codes, and how to put it all together.

7.3.1 What is a federated content component?

A federated content component is a component used to retrieve and store information from a federated data source, such as Content Manager. Working with Content Manager data, a federated content component stores a reference to a Content Manager document in LWWCM. From LWWCM, users can perform searches for federated content and add the results of the search into the LWWCM Web pages. These searches can be for content or for specific federated attributes.

Using federated content components, you can:

- ▶ Search Content Manager objects using a federated search template and select a reference to it to be stored in the content.
- ▶ Store federated content component searches as reusable library components.
- ▶ Add federated content components to content templates.
- ▶ Clear Content Manager content reference from the component and select a new reference.
- ▶ View attributes of the Content Manager content reference.
- ▶ Preview the Content Manager content.

7.3.2 Why use federated content components?

Using federated content components enables users to have access to multiple sources of information. This means that a user can use one query to search against multiple sources to get a result set. There is no need for the user to understand how and where the information is stored.

7.3.3 Setting up a federated content component

To use a federated content component, you need to configure the system to be capable of performing federated searches. In addition, federated entity and search templates must be defined in the Information Integrator for Content Administration Client. Once they are defined, we need to create the federated content component (either in a component library or within a content object) and include it in the Web page content.

Prerequisites and configuration

The following is a list of tips and prerequisites for using the federated content components:

- ▶ Information Integrator for Content V8.2 with Fix Pack 3 or greater must be installed on your federated data server.
- ▶ Check your Information Integrator for Content configuration and make sure that you have the *cmbds.ini* file under the <CMGMT_HOME> directory. Refer to Chapter 5, “Before you begin integration: pre-integration validation” on page 63 for more detailed information on the setup.
- ▶ To link to a federated data source, the following parameters must be edited within the *Connector* section of the *connect.cfg* file.
 - DefaultServerAlias: alias name created when the Information Integrator for Content is installed.
 - DefaultUsername: default user ID used to log on to the federated database.
 - DefaultPassword: password for the user ID above.
 - MaxResultsReturned: maximum number of results returned by the search query.

Example 7-1 shows a code snippet of the *connect.cfg* file with these parameters.

Example 7-1 connect.cfg: code snippet

```
<FederatedConnector>
  <DefaultServerAlias value = "ICMNLSDb" />
  <DefaultUsername value = "icmadmin" />
```



```
<DefaultPassword value = "password" />
<MaxResultsReturned value = 100 />
</FederatedConnector>
```

- Make sure proper user IDs are set up in LWWCM and Information Integrator for Content for accessing federated data sources.

Note: If you do not specify a default user name and password, LWWCM will try to log on to the Content Manager's federated data source with the current LWWCM user ID. If this user ID does not have access rights to the federated data source, then the user cannot create the federated component.

What we need from Information Integrator for Content

In order to make federated content components work, we need to have the *resource back-end server(s)*, the corresponding *federated entities* and *search templates* defined in the Information Integrator for Content server.

Defining the resource back-end server (Content Manager V8 server)

You must define the resource back-end server, in this case, the Content Manager server, before you connect to it.

Note: Before you define the server, make sure that the corresponding server connector is installed during the Information Integrator for Content installation.

To define the Content Manager V8 server within Information Integrator for Content, follow these steps:

1. Start Information Integrator for Content administration console by selecting **Start -> Programs -> Enterprise Information Portal for Multiplatforms 8.2 -> Administration**.
2. Log in with a proper user ID and password.
3. Right-click **Servers** and select **New -> Content Manager V8** from the context menu. See Figure 7-7 on page 126.

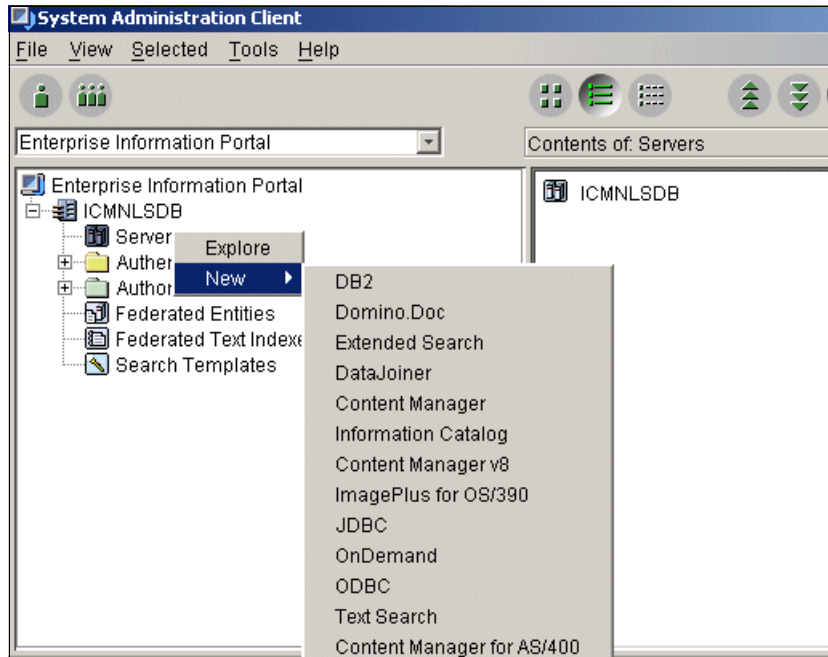


Figure 7-7 Define new resource back-end server: Content Manager V8 server

4. Enter a new server name. Click **Test connection**. See Figure 7-8.

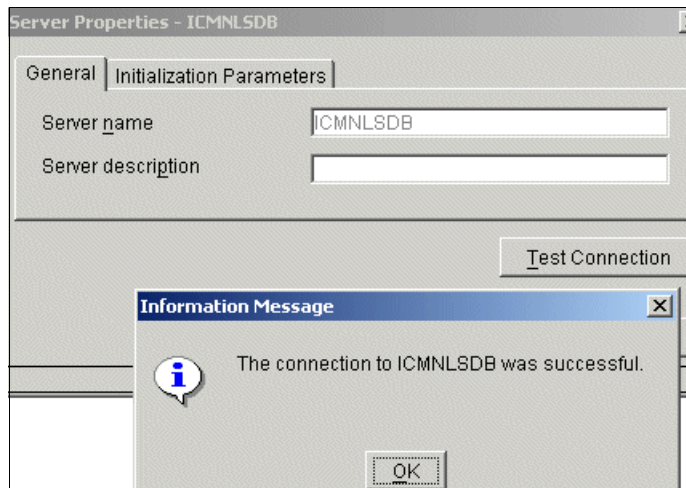


Figure 7-8 Test connection to the newly defined content server

Information Integrator for Content logs on to the content server using the user ID and password you entered when starting the system administration client. If this fails, it prompts you to enter a valid user ID and password.

5. Click **OK**.

Refreshing the server inventory

You must refresh the inventory of the server you defined earlier. If the server already exists in your Information Integrator for Content, you should still refresh its inventory in order to map the federated attributes with the item type attributes in Content Manager. If you forget to refresh the server inventory, you will not see any (first-time creation) of the latest server inventory item type attributes during mapping.

1. Go back to the Information Integrator for Content (or EIP) System Administration Client window.
2. Right-click the new content server you defined and select **Refresh Server Inventory**. See Figure 7-9.

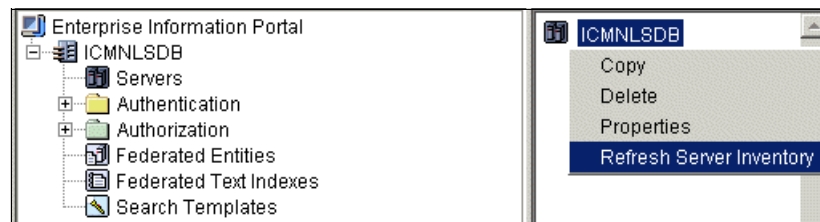


Figure 7-9 Refresh server inventory for the newly defined content server

Creating a federated entity

To create a new federated entity, follow these steps:

1. Start Information Integrator for Content administration console by selecting **Start -> Programs -> Enterprise Information Portal for Multiplatforms 8.2 -> Administration**.
2. Log in with a proper user ID and password.
3. Right-click **Federated Entities** and select **New -> Wizard** from the context menu.
4. Enter the name of the new federated entity and click **Next**.
5. Add the names of the federated attributes:
 - a. For each federated attribute, enter a name and click **Add**.
 - b. When all are added, click **Next**.

6. Map the federated attributes with the attributes (of item type InsuranceType) in Content Manager:
 - a. To map, click **Retrieve Server Inventory**. The list of the content server attributes per item type will appear in the right panel.
 - b. Select the federated attribute from the top drop-down list, then select the proper match attribute in the content server on the right panel, and click **Map**.
 - c. Repeat the previous step for each federated attribute. Go through each attribute from the top drop-down list.
 - d. When finished mapping the attributes, click **Next**.
7. If necessary, set properties such as length and data type for each federated attribute. Click **Next**, and then click **Finish** to confirm the setup.

Figure 7-10 shows the federated attributes' (entities) mappings. The native attributes represent the attributes in the Content Manager server.

Federated entity mappings		
Federated ...	Native attri...	Native ent
InName	IT_Name	InsuranceT
InDesc	IT_Descript...	InsuranceT
InNumber	IT_Number	InsuranceT
InGroup	IT_TargetG...	InsuranceT

Figure 7-10 Mapped attributes for the insurance case study

Creating a federated search template

Once the federated entity is created, we need to create a new search template to be used by the LWWCM federated content component. To create a new search template, follow these steps:

1. Start the Information Integrator for Content administration console by selecting **Start -> Programs -> Enterprise Information Portal for Multiplatforms 8.2 -> Administration**.
2. Log in with a proper user ID and password.
3. Right-click **Search Templates** and select **New -> Wizard** from the context menu.
4. Enter a name of the new search template, select **Federated entity**, and click **Next**.
5. Add searchable federated attributes in the search template. For each federated attribute that needs to be in the search template, do the following:

- a. Enter a name for the Search criterion.
 - b. Select an attribute from the Federated attribute list.
 - c. Select a default operator. Optionally, click **Select all**.
 - d. Click **Save the Current Criterion**.
6. The default option is Search using all criteria (AND). This can be modified. In our scenario, we select the Search using any criteria (OR) option. Click **Next**.
 7. Click **Next**.
 8. Select **Users/Groups** from the left panel and click **Add**.
 9. Click **Next** and then click **Finish**.

Creating a new federated content component in LWWCM

There are two different approaches for the creation of the federated content components (federated components):

- ▶ Creating a federated component *within the component library* and using it as a query search against the Content Manager repository.
- ▶ Creating a federated component *directly within a content object* and using it to display the results within a content page.

Using the first approach, the federated component can be shared and used in other content pages.

Creating a federated component in the component library

To create a federated component in the component library, follow these steps:

1. Open the LWWCM administration client.
2. Expand **Design & Development**.
3. Select **Component Library** and click **New**.
4. In the dialog box, select **Federated Content Component** and click **OK**.
5. Enter a name for the component.
6. Select the **Content Reference** tab.
7. Click **Select Reference** (see Figure 7-11 on page 130). From the list, choose a search template (previously created in this section) or a previously created query.

Figure 7-11 shows the 'Select Reference' dialog box. The 'Search Template' is set to 'InsuranceType'. The 'Criteria' section contains the following fields:

Field	Operator	Value
InsuranceName	LIKE	%
InsuranceDesc	LIKE	%
InsuranceNumber	LIKE	%
InsuranceGroup	LIKE	%

Figure 7-11 Query with the Insurance Type search template

8. Any federated content matching the supplied search query will be in the result list (see Figure 7-12). Select one of the federated records from the result list by clicking any of its attributes.

Figure 7-12 shows the results of the search query. The results are displayed in a table with the following data:

InName	InDesc	InNumber	InGroup
Life Insurance	Life Insurance Detailed Description	00000001	Personal
Automobile Insurance	Automobile Insurance Detailed Description	00000002	Personal
Longterm Care	Longterm Care Insurance Detailed Description	00000003	Personal
Investment Mgmt	Investment Management Detailed Description	10000001	Business
Retirement Program	Retirement Program Insurance Detailed Description	10000002	Business
Insurance Name 1	Description of Insurance Name 1	11111111	Business
MyInsurance	personal insurance	98989	Personal

Figure 7-12 Query results

9. Click **Save** and then click **Close**.

You will notice that in the federated component form (see Figure 7-13 on page 131), under the Select Reference button, there are some other buttons:

- **Clear Reference:** removes the currently selected Federated Content from this Component.
- **Unlock Version:** by default, the Federated Content Components are *locked*. The Federated Content Component can be unlocked by clicking Unlock Version. If Unlocked, each time the Federated Content Component is rendered, the current version is retrieved from the Federated Data Source.

- **Lock:** locks the Federated Content Component to the current version. If the Federated Data Source is updated, the Federated Data source will not use the updated version, but will use the *locked* version.

Note: This does not refer to the Web Content Management *Version Control* feature. This only applies to the version control feature of the federated data source. In our case, it is Content Manager.

- **Refresh Version:** updates the locked federated content reference to the latest version stored in the federated data source. The status of the Federated Content Component is still considered to be locked.
- **Preview Content:** displays the selected federated resource.
- **View Attributes:** opens a dialog displaying the current attributes of the selected federated content.

Figure 7-13 Federated Component form

A global user ID and password can be used to access the federated content servers. This eliminates the need to provide access to all LWWCM users to the federated content server.

Creating a federated component within a content object

Creating federated components from the proper content is similar to the process we described earlier. The difference is the location where the component is created.

In our scenario, we assume that the content is already created and we want to add a federated component to it. To do so, proceed with the following steps:

1. Open the LWWCM administration client.
2. Select **Content Management -> Content Library**.
3. From the right panel, select a content object and click **Edit**.
4. Click **Component Manager**.
5. Click **Add**.
6. Enter a name for the new component to be added.
7. Select **Federated Content Component** from the Type list.
8. Click **OK**, and then click **OK** again.
9. Select the **Contents** tab.
10. There will be a label with the name of the recently added federated component (see Figure 7-13 on page 131); click **Select Reference**.

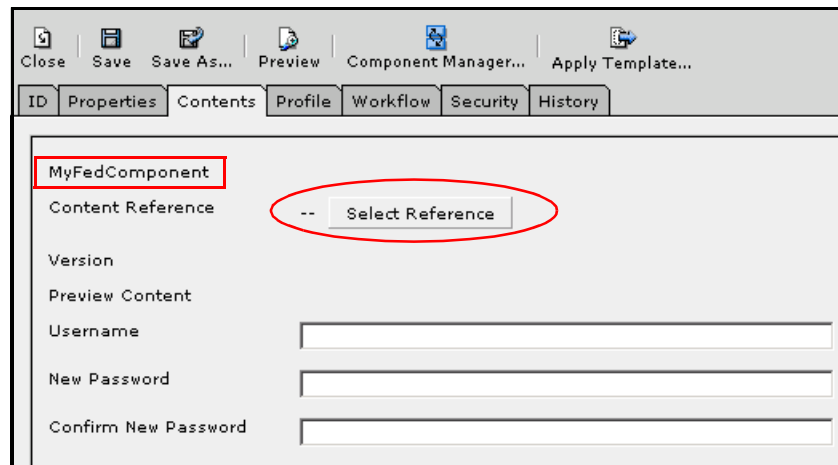


Figure 7-14 Federated component added within a LWWCM content page

11. Enter a search template and the search criteria for each attribute (the default value to obtain all is the % character)
12. Click **Search**.
13. Select one referenced source object.
14. Click **Save**, and then click **Close**.

Once we have included the federated data within the selected content object page, we need to modify the appropriate *Page Design* so that the federated content can be shown within the appropriate Web page.

Using component tags within the page design

A federated component can be used in a component library or in the context of a content document. There are two tags used to reference the two types of federated components:

- ▶ FederatedLibCmpnt tag
- ▶ FederatedCmpnt tag

Using the federated component, you can also pull the *attributes* of the Content Manager content or the content *document* itself into a Web page. You can either display a link to the attached content or render the Content Manager content directly into the page if the MIME type is of type text/* . Specifically, the three approaches are:

- ▶ Rendering federated attributes
- ▶ Rendering federated content links
- ▶ Rendering federated content

FederatedLibCmpnt tag

When used in the *component library*, it can be referenced in the page design as follows:

```
<FederatedLibCmpnt name="" scope="" attributename="" formatting="" start="" end="" />
```

Table 7-2 describes the available parameters for this tag.

Table 7-2 Available parameters for FederetadLibCmpnt

Parameter	Description
Name	The name of the library component.
Scope	Defines what information to display from the federated content reference. Valid options includes: content, contentURL and attribute.
Attributename	Sets the name of the federated attribute to display. It is only set when Scope is set to attribute.
Formatting	For attributes of type date, it can be used to format the retrieved value. Valid options for this tag attribute are outlined in the javadoc pages for class SimpleDateFormat.
Start and End	These tags behave in the same way as the other LWWCM tags. They can be used to create links to content when the Scope parameter is set to contentURL.

Note: The parameters `scope= "content"` and `scope="contenturl"` are only supported for the Content Manager repository. For other repositories, you should only use `scope="attribute"`.

FederatedCmpnt tag

When used in the *context of content*, it can be referenced in the page design as follows:

```
<FederatedCmpnt type="" context="" name="" key="" scope="" attributename=""  
formatting="" start="" end="" />
```

Table 7-3 describes the available parameters for this tag.

Table 7-3 Available parameters for FederetadCmpnt

Parameter	Description
Type	Represents the type of an object the inline component is in. Valid options include: content, site, sitearea.
Context	Indicates the context of the type of an object the inline component is in. Valid options include: current, selected and autofill.
Name	Represents the name of an object the inline federated component is in. It is only set if Context is set to selected.
Key	The name given to the federated content inline component in the object.
Scope	Defines what information to display from the federated content reference. Valid options include: content, contentURL, and attribute.
Attributename	Sets the name of the federated attribute to display. It is only set when Scope is set to attribute.
Formatting	For attributes of type date, this can be used to format the retrieved value. Valid options for this tag attribute are outlined in the javadoc pages for class SimpleDateFormat.
Start and End	These tags behave in the same way as the other LWWCM tags. They can be used to create links to content when the Scope parameter is set to contentURL.

Rendering federated attributes

Federated attributes of the MIME type text/* can be rendered directly into the Web page. Retrieval of BLOB attributes is not supported. As an example, we render all the available attributes of the life insurance type into our insurance Web page. The federated component needs to be added to the Life Insurance

content in LWWCM. The component needs to be configured to reference the Life Insurance content in Content Manager. After this, we also need to add the four tags into the page design (Insurance Layout). Example 7-2 shows the four tags used to render Content Manager attributes to the Web page.

Example 7-2 Rendering Content Manager attributes into a Web page

```
<b>Name:</b> <FederatedCmpnt AttributeName="InName" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Description:</b> <FederatedCmpnt AttributeName="InDesc" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Number:</b> <FederatedCmpnt AttributeName="InNumber" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Group:</b> <FederatedCmpnt AttributeName="InGroup" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>
```

If you want to do the same with federated components stored in the component library, Example 7-3 shows what your tags would look like.

Example 7-3 Rendering CM attr in Web using stored federated component

```
<b>Name:</b> <FederatedLibCmpnt name="Life Insurance" Scope="attribute"
AttributeName="InName"/><br>

<b>Description:</b> <FederatedLibCmpnt name="Life Insurance" Scope="attribute"
AttributeName="InDesc"/><br>

<b>Number:</b> <FederatedLibCmpnt name="Life Insurance" Scope="attribute"
AttributeName="InNumber"/><br>

<b>Group:</b> <FederatedLibCmpnt name="Life Insurance" Scope="attribute"
AttributeName="InGroup"/><br>
```

Note that FedCmpnt in Example 7-2 on page 118 is the name of our federated inline component and Life Insurance in Example 7-3 on page 120 is the name of our federated library component.

Rendering federated content links

When we created the life insurance content in Content Manager, we also attached a text document with a short description. The code shown in Example 7-4 on page 136 generates a line showing the URL and a link on the life insurance Web page. Clicking this link will open the text document. When the link is clicked, the FederatedProxy module of LWWCM will go to Content Manager and fetch the content.

Example 7-4 Rendering Content Manager content link into Web page

```
<b>Link:</b> <FederatedCmpnt context="current" key="FedCmpnt"
Scope="contentUrl" type="content"/><br>
<b>Insurance URL:</b> <FederatedCmpnt context="current" end="'">";"
key="FedCmpnt" Scope="contentUrl" start="&lt;a href="'" type="content"/>Click
here open Life Insurance.txt</a><br>
```

If you want to do the same with the federated components stored in the component library, Example 7-5 shows what your tags would look like.

Example 7-5 Rendering CM attr in Web using stored federated component

```
<b>Link:</b> <FederatedLibCmpnt name="Life Insurance" Scope="contentUrl"/><br>
<b>Insurance URL:</b> <FederatedLibCmpnt name="Life Insurance" end="'">";"
Scope="contentUrl" start="&lt;a href="'" type="content"/>Click here open Life
Insurance.txt</a><br>
```

Rendering federated content

Only Content Manager content of MIME type text/* can be rendered directly into a Web page. To achieve this, you would have to use the tag as shown in Example 7-6 in your page design.

Example 7-6 Rendering Content Manager content into a Web page

```
<b>Content:</b> <br><FederatedCmpnt context="current" key="FedCmpnt"
Scope="content" type="content"/>
```

If you want to do the same with federated components stored in the component library, Example 7-7 shows what your tags would look like.

Example 7-7 Rendering CM attr in Web using stored federated component

```
<b>Content:</b> <br><FederatedLibCmpnt name="Life Insurance"
Scope="content"/>
```

For more detailed information regarding federated content component, refer to the LWWCM online Information Center.

7.3.4 Putting it all together

In this section, we describe the required steps to include all the features described in the earlier section and obtain a Web page in which the content comes from Content Manager, rendered by the federated components. We will continue with the insurance case study for our demonstration. The result of putting it all together is shown in Figure 7-15 on page 138.

- ▶ Ensure that you have created the following from Content Manager:
 - Insurance attributes: IT_Number; IT_Name; IT_Description; IT_TargetGroup
 - Insurance item type: InsuranceType
 - Sample content, for example, imported text documents.

When you deploy a search, selecting **InsuranceType** as the item type from Content Manager, you will obtain a list of the records similar to that of the top search result table shown in Figure 7-15 on page 138. The selected entry shown in the figure (*Life Insurance*) has a text file attached which describes the detailed information about the insurance plan.

- ▶ Modify the *Insurance layout* page design to include the proper tags to show the attributes. See Example 7-2 on page 135 as an example.
- ▶ Create a new content page or modify from an existing one. In the Content tab, add a federated component, and select the **Life Insurance** record as described in “Creating a federated component within a content object” on page 131.

The result of these combined steps is shown in Figure 7-14 on page 132, where the content of the Web page is retrieved from Content Manager using the federated components.

**Search result
from CM client
for Windows**


Folder: <Search Results>

IT_Type	IT_Number	IT_Name	IT_Description	IT_Target...
	00000002	Automobile Ins...	Automobile Insurance Detailed Des...	Personal
	00000003	Longterm Care	Longterm Care Insurance Detailed ...	Personal
	10000001	Investment M...	Investment Management Detailed ...	Business
	10000002	Retirement Pr...	Retirement Program Insurance Det...	Business
	00000001	Life Insurance	All about life insurances version 5	Person

5 Item(s) 1 Selected Total 1 Selected in "InsuranceType"

- > About
- > News & Events
- > Strategy & Reports
- > Admin & HR
- > Sales & Marketing
- > Insurance Products
 - Overview
 - Personal and Families
 - Life Insurance
 - Life-term Care Insurance
 - Automobile Insurance
 - Companies

Home > Insurance Products > Personal and Families > Life Insurance



Name: Life Insurance
Description: All about life insurances version 5
Number: 00000001
Group: Person

Content:
Life Insurance is the key of your family's financial security. It protects your financial resources against the uncertainties of life so you can protect your family's future. But planning for our own death is not something that many of us want to do. Having support in place when the unfortunate happens can help those left behind take care of arrangements and recover from the loss easier. Life insurance can be an important part of that support.

URL: Click here for more detailed information

Document: <undefined>

Life Insurance is the key of your family's financial resources against the uncertainties of family's future. But planning for our own death is not something that many of us want to do. Having support in place when the unfortunate happens can help those left behind take care of arrangements and recover from the loss easier. Life insurance can be an important part of that support.

**Attached file
in the CM
content
opened for
preview**

Figure 7-15 Resulting Web page of publishing CM content using federated components

7.4 Connect tags

There are three ways of retrieving data from Content Manager and publishing the data to LWWCM Web sites. One way is by using the connect tags. In this section, we discuss what the connect tags are, why we use them, how to set them up with sample codes, and how to put them all together.

7.4.1 What are connect tags?

Connect tags are tags that are embedded in HTML documents and which you can use in page design or component designs. They are used to retrieve data from external resources such as databases or Web sites. The connect tags in an HTML document specify the location in the document where the retrieved data should be placed and the layout of the retrieved data.

Using the connect tags, you can:

- ▶ Display data from external databases.
- ▶ Display other Web pages.
- ▶ Publish data from Content Manager.

7.4.2 Setting up connect tags

In this section, we show you how to set up, configure, and use the connect tags.

Prerequisites and configuration

Verify that the specific CM tag is in the connect.cfg file. See the bold text in Example 7-8. If not, you should add it to the file.

Example 7-8 Prerequisite tag in connect.cfg file

```
<DataProviders>
  <SQL
value="com.presence.connect.business.template.provider.SQLDataProvider" />
  <CM
value="com.presence.connect.business.template.provider.CM8DataProvider" />
</DataProviders>
```

Depending on the version of LWWCM you are using, sometimes, by default, this tag already exists in the connect.cfg file.

To be able to process the connect tag, the **Process Connect Tag** check box must be selected in *each page design* you use to show your content. See Figure 7-16 on page 140 for an example of the page design.

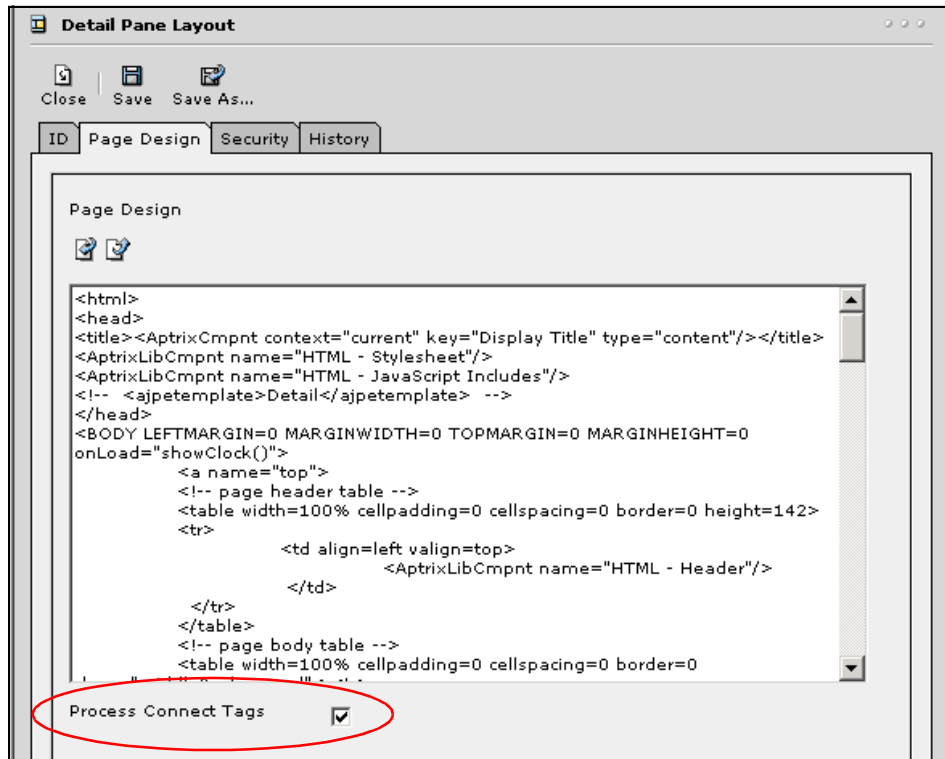


Figure 7-16 Process connect tags

When you insert your connect tag in a component where you have the option of Rich Text or HTML (similar to that in the Rich Text component), be sure that you choose **HTML**; otherwise, the connect tag will be not rendered. See Figure 7-17.

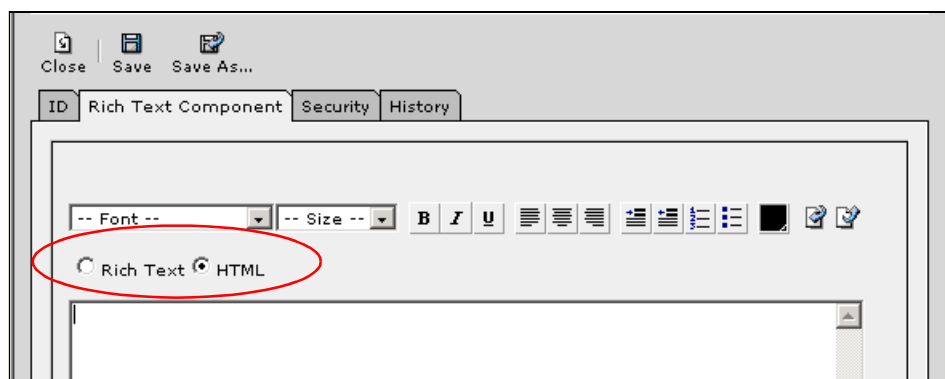


Figure 7-17 HTML option in Rich Text component

Using connect tags within the page design

The connect tag syntax is as follows:

```
<Connect MOD="" SRV="" {<Additional parameters>=""}>{Failure text}</Connect>
```

Table 7-4 describes the available parameters and options for this tag.

Table 7-4 Available parameters and options for the connect tag

Parameter	Description
MOD	Represents the module used to display data. Valid options include: <ul style="list-style-type: none">▶ TEMPLATE - Used when calling data from external database.▶ WEB - Used when displaying data from another Web page.▶ AGGREGATOR - Used when displaying data from part of another Web page.
SRV	Indicates the data provider for the displaying data. Valid options include: <ul style="list-style-type: none">▶ SQL (Only applicable when MOD="TEMPLATE")▶ LDAP (Only applicable when MOD="TEMPLATE")▶ CM (Only applicable when MOD="TEMPLATE")▶ HTML (Only applicable when MOD="WEB" or "AGGREGATOR")
Additional parameters	Additional parameters associated with MOD and SRV. For example, if using MOD="TEMPLATE" SRV="SQL", additional parameters include DB and SQL.
{Failure text}	Optional. If LWWCM failed to retrieve the content, the "Failure text" content will be displayed on the Web page. The failure text can be any valid connect or HTML tags.

As mentioned earlier, using the connect tags, we can:

- ▶ Display external data
- ▶ Display other Web pages
- ▶ Publish data from Content Manager

We examine how to implement the tags with a closer look at the syntax involved.

Using connect tags to display external data

There are three types of external databases that the connect tags can access to retrieve their data and display it on a Web page. They include:

- ▶ SQL databases
- ▶ LDAP directory
- ▶ Content Manager

Retrieving data from the SQL database

The syntax used in the connect tag to retrieve data from SQL database is:

```
<CONNECT MOD="TEMPLATE" SRV="SQL" DB="" USER="" PWD="" SQL=""> {Failure  
text}</CONNECT>
```

Where:

- ▶ MOD="TEMPLATE": this module is used when working with external databases.
- ▶ SRV="SQL": indicates the data provider is an SQL database.
- ▶ DB: specifies the SQL database server name and the protocol.
- ▶ SQL: the SQL statement used to retrieve the actual data.
- ▶ USER: optional. Specifies the user ID to access the database.
- ▶ PWD: optional. The password for the database user.
- ▶ TEMPLATE: optional. You can define a formatter file for formatting what and how you want to see as far as the result on your page. If you do not specify this parameter, the retrieved data will be displayed with minimum formatting.

Example 7-9 shows an example of using the connect tag to retrieve data from an SQL database.

Example 7-9 Retrieving data from SQL database

```
<CONNECT MOD="TEMPLATE" SRV="SQL" DB="jdbc:odbc:sample" SQL="Select * from  
customer" USER="icmnlbdb" PWD="password" > Failed to retrieve data from the SQL  
database </CONNECT>
```

Retrieving data from the LDAP directory

Lightweight Directory Access Protocol (LDAP) such as the IBM Directory server can be accessed using the connect tag.

The syntax used in the connect tag to retrieve data from LDAP directory is:

```
<CONNECT MOD="TEMPLATE" SRV="LDAP" HOST="" USER="" PWD=""> {Failure  
text}</CONNECT>
```

Where:

- ▶ MOD="TEMPLATE": this module is used when working with external databases.
- ▶ SRV="LDAP": indicates the data provider is an LDAP directory.
- ▶ HOST: the host ID (IP address or domain name DNS) of the LDAP server.
- ▶ USER: the user's distinguished name.
- ▶ PWD: the user password.

Example 7-10 shows an example of using the connect tag to retrieve data from an LDAP directory.

Example 7-10 Retrieving data from LDAP directory

```
<CONNECT MOD="TEMPLATE" SRV="LDAP" HOST=itso.ibm.com USER=ldapadmin  
PWD=password> Failure to retrieve data from the LDAP directory </connect>
```

Retrieving data from the Content Manager server

The syntax used in the connect tag to retrieve data from the Content Manager server is:

```
<CONNECT MOD="TEMPLATE" SRV="CM" SERVER="" USER="" PWD="" SEARCH=""  
TEMPLATE="" > {Failure text}</CONNECT>
```

Where:

- ▶ MOD="TEMPLATE": this module is used when working with external databases.
- ▶ SRV="CM": indicates the data provider is the Content Manager server.
- ▶ SERVER: the Content Manager Library Server name.
- ▶ USER: optional. The user ID used to access the Content Manager server.
- ▶ PWD: optional. The user password.
- ▶ SEARCH: used to specify search query.
- ▶ TEMPLATE: optional. Used to format the retrieved data.

Example 7-11 shows an example of using the connect tag to retrieve data from Content Manager.

Example 7-11 Retrieving data from Content Manager

```
<CONNECT MOD="TEMPLATE" SRV="CM" SERVER="icmnlbdb" USER="icmadmin"  
PWD="password" SEARCH={search string} TEMPLATE={template filename} > Failed to  
connect to the Content Manager server data </CONNECT>
```

Note that the values for the SEARCH and TEMPLATE parameters are not supplied. In later section of this chapter, we discuss in more detail of the parameters and on how to use the connect tag to retrieve data from Content Manager.

Using connect tags to display data from the Web

There are two options to retrieve data from an existing Web page and to display it in the LWWCM Web site:

- ▶ Retrieving complete page from an existing Web site
- ▶ Retrieving part of a page from an existing Web site

Retrieving complete page from an existing Web site

The syntax used in the connect tag to retrieve an existing Web page is:

```
<Connect MOD="WEB" SRV="HTML" ACTION=""> {Failure text}</connect>
```

Where:

- ▶ MOD="WEB": this module is used when retrieving data from a Web page.
- ▶ SRV="HTML": indicates the data provider is HTML.
- ▶ ACTION: Web site that you want to obtain information from.

The data retrieved from the connect tag will replace the tag. The HTML header will be added to the header of the original HTML file.

Example 7-12 shows an example of using the connect tags to retrieve data from a Web site.

Example 7-12 Retrieving complete page data from an existing Web site

```
<Connect MOD="WEB" SRV="HTML" ACTION="http://www.xyz.com"> {Failure  
text}</connect>
```

Retrieving a partial page from an existing Web site

If you do not want to retrieve the entire Web page, but instead merge a section of the Web page, you have to use the AGGREGATOR module instead of the WEB module. The syntax used in the connect tag to retrieve an existing Web page is:

```
<Connect MOD="AGGREGATOR" SRV="HTML" ACTION="" START="<table>"  
END="</table>"> {Failure text}</connect>
```

Where:

- ▶ MOD="AGGREGATOR": this module is used when retrieving partial Web page.
- ▶ SRV="HTML": indicates the data provider is HTML.
- ▶ ACTION: Web site that you want to obtain information from.
- ▶ START: the string where you want to start stripping an Web page content.
- ▶ END: the string where you want to stop stripping an Web page content.

Important: The START and END parameter tags are *case-sensitive*. You can use any text from the Web site; however, if you use the HTML tags, you must specify the complete tag (for example, use <table> and not just <tab>) and not an HTML tag with attributes ().

Example 7-13 on page 145 shows an example of using the connect tag to retrieve partial page data from an existing Web site.

Example 7-13 Retrieving partial page data from an existing Web site

```
<CONNECT END="Python app." START="Develop Python"  
ACTION="http://www-136.ibm.com/developerworks/" SRV="HTML"  
MOD="AGGREGATOR"></CONNECT>
```

The source Web site, IBM developerWorks®, is shown in Figure 7-18. We are interested in displaying the article summary related to developing Python. We create new content which contains the connect tag as shown in Figure 7-19 on page 146 (the same as shown in Example 7-13). The connect tag strips part of the Web page starting from “Develop Python” and ending with “Python app.”. Figure 7-20 on page 146 shows the end result: the retrieved partial page data from the developerWorks Web site is shown in our case study Insurance Web site.

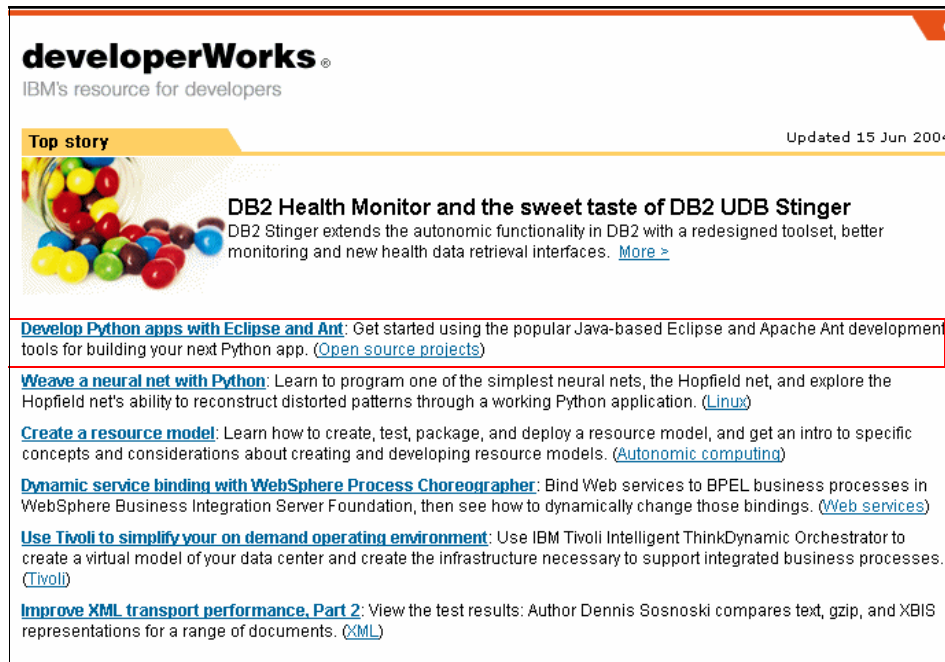


Figure 7-18 IBM developerWorks Web site

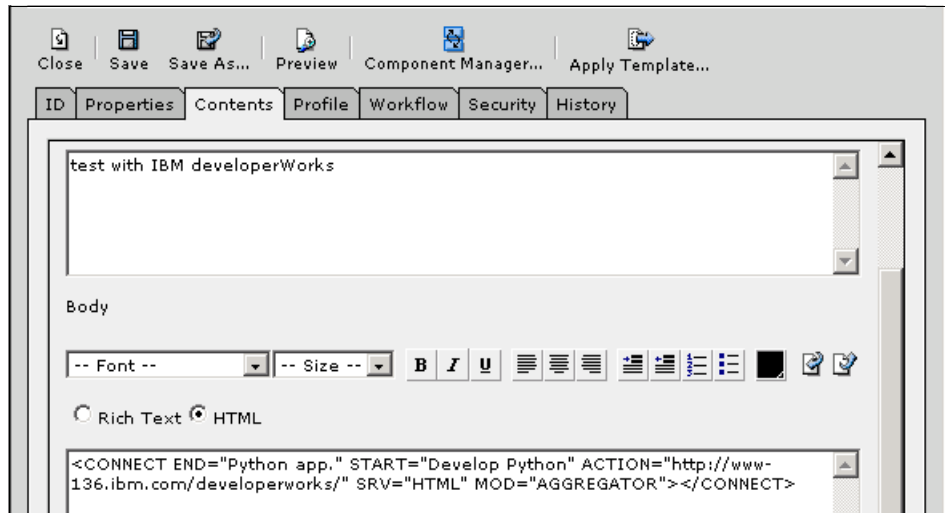


Figure 7-19 Connect tag to strip partial Web page"

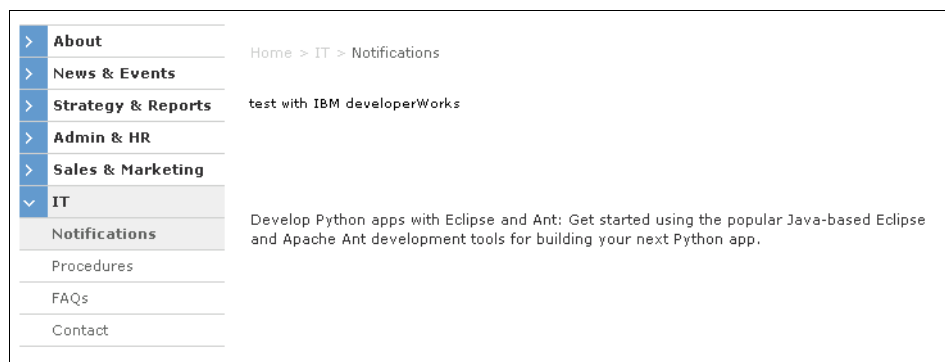


Figure 7-20 Result of stripping partial Web page content to Insurance Web site

Using connect tags to cache LWWCM components

Components placed in a page design can be individually cached by replacing the component references within a connect tag. The connect tag uses the renderer module to cache the LWWCM component. This module is defined in the connect.cfg file and, by default, is enabled.

We recommend creating a new module in the connect.cfg file, for example, ComponentRenderer, that points to the same class used for the default renderer. This way, if we later disable the renderer module, our newly created module will still be in effect: the newly created module will be independent of the default setup.

Example 7-14 shows the additional code (in bold text) that we need to add to create this new renderer in the connect.cfg file.

Example 7-14 Copy default renderer module in connect.cfg

```
<!-- Renderer Module -->
<Default
class=com.ibm.workplace.Connect.app.render.RendererModule
remoteAccess=true autoLoad=false />

<ComponentRenderer
class=com.ibm.workplace.Connect.app.render.RendererModule
remoteAccess=true autoLoad=false />
```

Once the new ComponentRenderer is created in the connect.cfg file, we can use it in the connect tag. The syntax is as follows:

```
<Connect MOD="ComponentRenderer" SRV="cmpnt" PATH="" SOURCE="" CMPNTNAME=""
CACHE="" EXPIRES=""> {Failure text}</connect>
```

Where:

- ▶ MOD="ComponentRenderer": the name of the renderer module.
- ▶ SRV="cmpnt": indicates the data provider is the LWWCM component.
- ▶ PATH: sets the context for the component.
- ▶ SOURCE: indicates we are caching from the component library.
- ▶ CMPNTNAME: specifies the name of the component.
- ▶ CACHE: indicates whether to cache the site or session.
- ▶ EXPIRES: Indicates when the cache will expire, in seconds. After the expiration time, the component will be rendered again.

An example of using the connect tag to cache a LWWCM component is shown in Example 7-15.

Example 7-15 Using connect tag to cache Web Content Manager component

```
<Connect MOD="ComponentRenderer" SRV="cmpnt" PATH="/Site/SiteArea/Content"
SOURCE="library" CMPNTNAME="Nav1" CACHE="site" EXPIRES="REL 7200s"> </connect>
```

In the above example, the navigator component, Nav1, will be rendered for the first time, put in the cache and used from the cache until after two hours have elapsed. To disable caching, we can set CACHE to none. See Example 7-16 on page 148.

Example 7-16 Disable component renderer caching

```
<Connect MOD="ComponentRenderer" SRV="cmpnt" PATH="/Site/SiteArea/Content"
SOURCE="library" CMPNTNAME="Nav1" CACHE=none> </connect>
```

Note: Custom caching cannot be used if the basic caching is used as your server's default Web content cache.

For more detailed information regarding connect tags, refer to the LWWCM online Information Center.

7.4.3 Putting it all together

In the previous section, we described many ways of using the connect tags to display content from external sources. In this section, we work exclusively with Content Manager data, and describe the required steps to publish Content Manager data using the connect tags.

As mentioned earlier in “Retrieving data from the Content Manager server” on page 143, LWWCM's connect tags enable us to retrieve data from Content Manager server and display it on our Web site.

We continue with the insurance case study for our demonstration.

The syntax used in the connect tag to retrieve data from Content Manager server is:

```
<CONNECT MOD="TEMPLATE" SRV="CM" SERVER="" USER="" PWD="" SEARCH=""
TEMPLATE="" > {Failure text}</CONNECT>
```

1. We need to specify the Content Manager Library Server name, user ID, and the associated password:

```
SERVER="icm1sdb" USER="icmadmin" PWD="password"
```

Make sure the Library Server you want to access is defined in the cmbicmsrvs.ini file of the host where LWWCM and WebSphere are installed.

2. We need to define a search string. In our example, we use the XPath query to search Content Manager Library Server for any InsuranceType items that have names (IT_Name) equal to Automobile Insurance and that have a target group (IT_TargetGroup) equal to Personal, and sort by insurance number (IT_Number). See the query string below:

```
SEARCH='/InsuranceType[@IT_Name="Automobile Insurance" AND
@IT_TargetGroup="Personal"] SORTBY (@IT_Number ASCENDING)'
```


For more information on how to perform a search in Content Manager, see Chapter 7, “Query language” in *Content Manager Implementation and Migration Cookbook*, SG24-7051.

3. Set up the formatter file. This file is written in HTML so you can use all the standard HTML tags. In addition, LWWCM provides result set tags which can be used to format retrieved data before it is returned to the browser. The retrieved data is in a set of rows. Using the result set tags, you can process each row, format the data, and display it to the browser.

The formatter file needs to be placed on the Web server. In our example, we place the formatter file, `template1.html`, in the directory:

`C:\IBM\WebSphere\AppServer\intalledApps\demo1\ilwwcm_ear\ilwwcm.war\templates\template1.html`

In the connector tag, we set the `TEMPLATE` as follows:

```
TEMPLATE="http://localhost:9080/ilwwcm/templates/template1.html"
```

Example 7-17 shows the code snippet for the formatter file, `template1.html`. LWWCM loops the result set, puts each result in a table row, where the `IT_NAME` attribute will be displayed in bold, with `IT_Description` on the next line. It also creates a link to the document part if it exists.

Example 7-17 Formatter file: `template1.html`

```
<TABLE>
  <RESULTSET >
    <LOOP>
      <TR>
        <TD>
          <B><FIELD NAME="IT_Name"> / <FIELD NAME="IT_Duration"></B>
          <BR/><FIELD NAME="IT_Description"> <A HREF="<FIELD NAME=Part>"
TARGET="_blank">...more</A>
        </TD>
      </TR>
    </LOOP>
  </RESULTSET >
</TABLE>
```

Putting it all together, the resulting connect tag for our case study is shown in Example 7-18.

Example 7-18 Connect tag to retrieve and display Content Manager data

```
<connect MOD="TEMPLATE" SRV="CM" SERVER="icm1sdb" USER="icmadmin" PWD="berlin"
SEARCH="/InsuranceType[@IT_Name="Automobile Insurance" AND
@IT_TargetGroup="Personal"] SORTBY (@IT_Number ASCENDING)'
TEMPLATE="http://localhost:9080/ilwwcm/templates/template1.html"></connect>
```

To use the connect tag, do the following:

1. Create new content.
2. Put the connect tag (as shown in Example 7-18 on page 149) in the Content tab as HTML. See Figure 7-21.
3. Save and look at the preview to see the result. See Figure 7-22 on page 151.

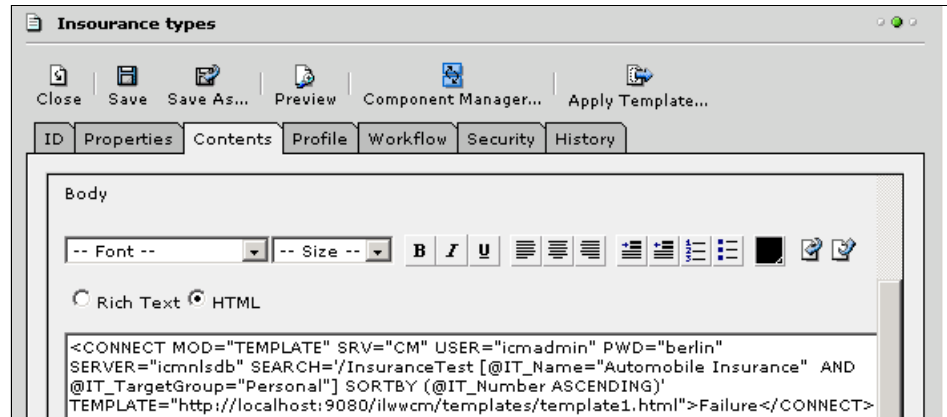


Figure 7-21 Create a new content with connect tag

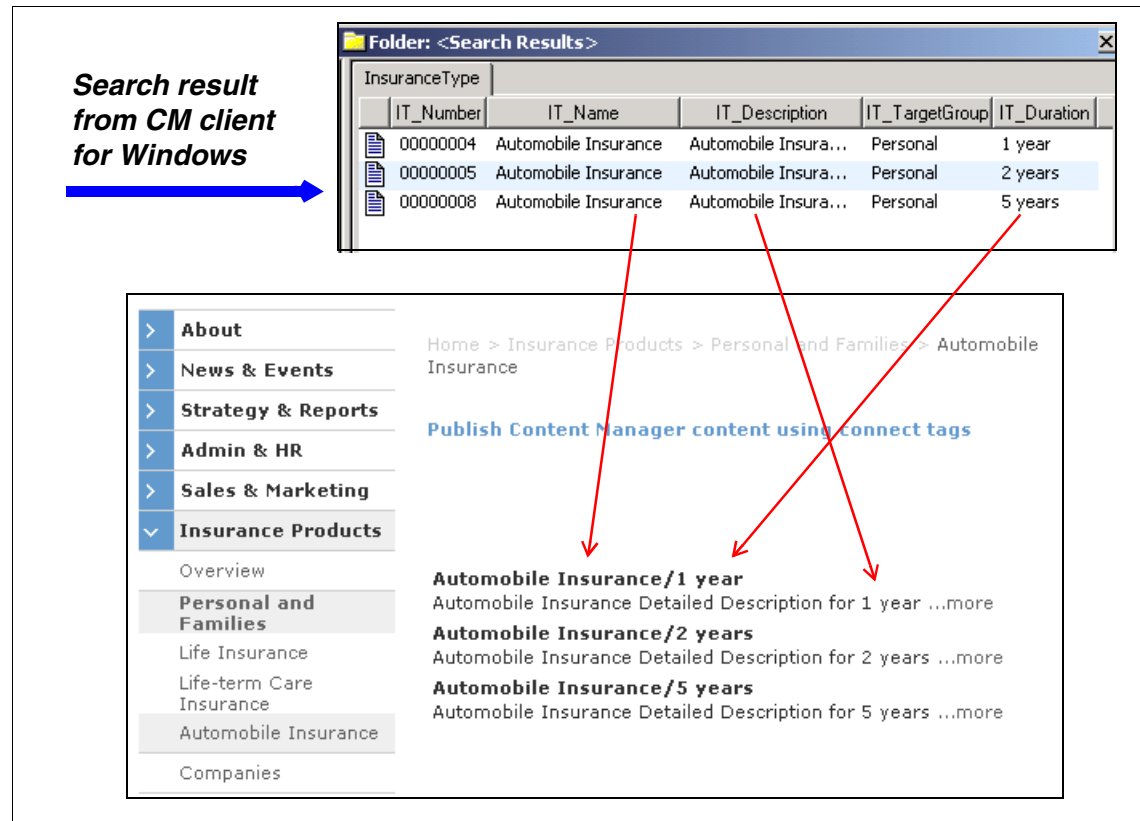


Figure 7-22 Resulting Web page of publishing CM content using connect tags

7.5 JSP component

One of the three ways of retrieving data from Content Manager and publishing the data to LWWCM Web sites is by using the JSP component. In this section, we discuss what it is, why we use it, how to set it up with sample codes, and how to put it all together.

7.5.1 What is a JSP component?

A *JSP component* is a component that provides the ability to embed JSPs directly into an LWWCM Web site. The JSP code is stored in a JSP file, which is referenced within a JSP component in LWWCM. The actual JSP file is stored in a local Web directory. Upon execution of the current LWWCM context, it is passed into the requested object.

Usually, the following needs to be done in such a JSP file:

- ▶ Reference LWWCM content or library components using <wcm> tags. This requires importing the wcm.tld tag library into the JSP file.
- ▶ Write the custom HTML and Java code. In this section, we deal exclusively with Content Manager Information Integrator so Content APIs can access Content Manager data. In general, it is potentially possible to access any back-end application data within LWWCM if the application provides Java APIs.

7.5.2 Why use the JSP component?

The JSP component allows users to write custom JSPs that are managed as LWWCM components. These components can be used to create additional functionality not provided by the out-of-the-box LWWCM.

This may be a preferred approach if Content Manager Java code is already available in your company. With some minor adjustments, the code can be reused to do its job in the context of LWWCM.

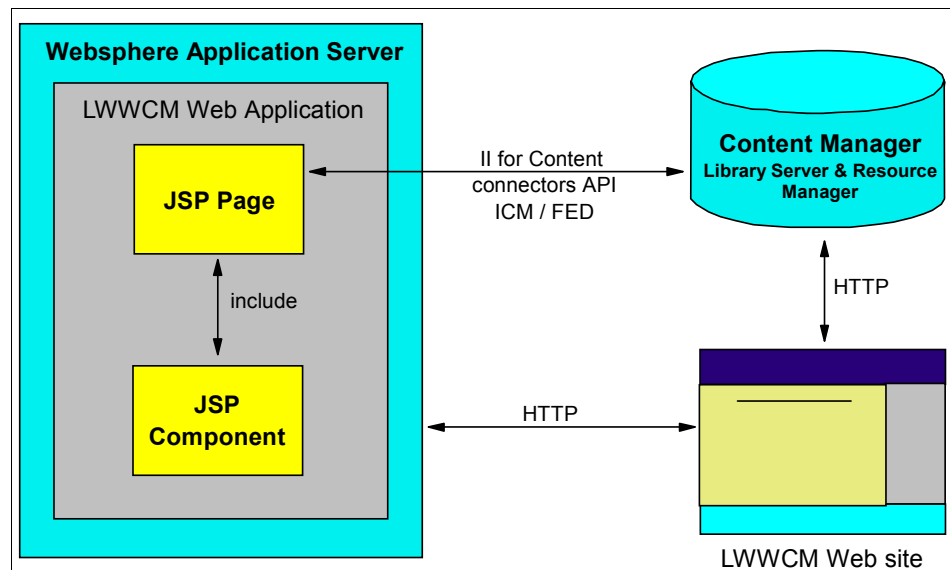


Figure 7-23 JSP component - CM - integration approach

7.5.3 Setting up a new JSP component

To use a JSP component, you need to write a JSP program and create and configure the JSP component to reference the corresponding JSP program.

Creation of a new JSP component

As for the federated content component creation, there are two approaches to the creation of a JSP component:

- ▶ Creating a JSP component *within the component library*
- ▶ Creating a JSP component *directly within a content object*

When creating a JSP component in the component library, it can be referenced by any page design and is independent of any other object shown on the page. When creating it within a content object, the component is tied to this content object and cannot be reused.

Creating a JSP component in the component library

To create a JSP component in the component library, follow these steps:

1. Open the LWWCM administration client.
2. Expand the **Design & Development**.
3. Select **Component Library**, and click **New**.
4. In the dialog box, select **JSP Component** and click **OK**.

Creating a JSP component within a content object

You can either put a JSP component directly into a content template so that it is available whenever new content is created with this template, or you can put the component manually into an existing content object.

To create a JSP component within a content object, follow these steps:

1. Within the LWWCM administration client, open an existing content object or create a new content object, open an existing content template or create a new content template.
2. Switch to edit mode if using an existing content object or template by clicking **Edit**.
3. Click **Component Manager**.
4. Click **Add**.
5. Enter a name for the new component.
6. Select **JSP Component** from the Type drop-down list and click **OK**.
7. Click **OK** again.
8. Click the tab **Contents** or **Content Template** and scroll to the bottom.

Configuration of the JSP component

The JSP component provides two text fields:

- **JSP Path:** This field allows you to enter the path of a local JSP file. The JSP file has to be local since the component internally performs a `RequestDispatcher.include(path)` and then merges the response into the resulting Web page. The limitation of the include method is that it is *local*. In most cases, you would put the JSP files into the same web-app folder as LWWCM. The path to be entered in this field has to begin with a forward slash and it is case sensitive. For example, assuming you created a JSP folder under the web-app folder and your JSP file is called `Test.jsp`, you must enter `/jsp/Test.jsp` for the JSP Path.

Tip: The web-app folder of LWWCM is by default located in the following path (The drive and folder names can differ from those in your installation):

```
[Drive]:\[Path]\WebSphere\AppServer\installedApps\[Node]\ilwwcm_war.ear\ilwwcm.war
```

- **Error Message:** This field allows you to enter an error message in case the JSP file is not available or the rendering of the JSP code is not successful. Having the error messages ensures that users of your Web site will not see cryptic Java error messages when an error is encountered.

Figure 7-24 on page 155 shows an example of the JSP component configuration.

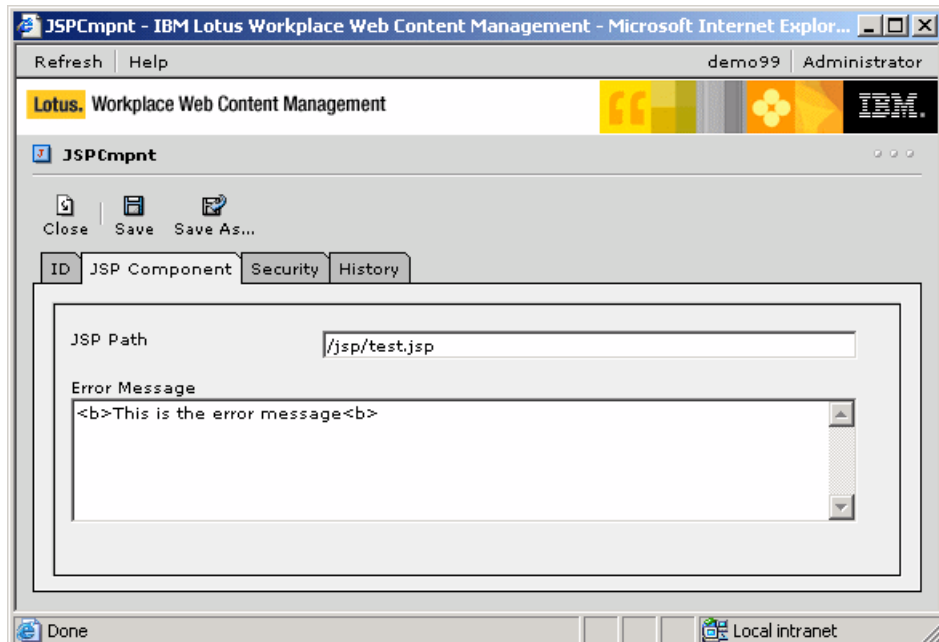


Figure 7-24 JSP component configuration

Before the component performs the “include,” it puts the rendering context (JSP_CMPNT_RENDERCONTEXT) into the request object so that the JSP can access it.

If you want to do something completely outside of the LWWCM Web application, then you would not use the JSP component. Instead, your JSP would become the front end in rendering, including performing authentication and setting up the rendering context. This JSP could then use the rendering tags to render content and components. This is addressed in “Sample code: using wcm JSP tags in JSP pages” on page 160.

Using the JSP component

As mentioned earlier, the JSP component can either be created in a component library or within the context of a content document.

When used in the component library, the JSP component can be referenced in a page design as follows:

```
<AptrixLibCmpnt name="JSPCmpnt"/>
```

When used in the context of content, it can be referenced in a page design as follows:

```
<ApatrixCmpnt context="current" key="JSP" type="content"/>
```

7.5.4 Sample code

In this section, we show some samples of using the JSP component to integrate with the external Content Manager content:

- ▶ “Sample JSP code: connection to Content Manager” on page 156 demonstrates how to connect to and disconnect from Content Manager in your JSP using the Information Integrator for Content connector technology. This code can act as a starting point for any JSP code connecting to Content Manager.
- ▶ “Sample JSP code: simple search” on page 158 demonstrates how to connect to a Content Manager server, execute a query and display the results using the ICM connector of Information Integrator for Content.
- ▶ “Sample code: using wcm JSP tags in JSP pages” on page 160 demonstrates the usage of wcm tags within JSP pages. This is not directly related to the integration with Content Manager but it is something very useful when creating JSPs that render LWWCM library components and content.

Sample JSP code: connection to Content Manager

Example 7-19 shows a sample JSP that demonstrates how to use the Content Manager Information Integrator for Content APIs to open and close a connection to a Content Manager system. Connecting to a Content Manager system is always the first step when integrating Content Manager content in LWWCM using the JSP component. For demonstration purposes, the JSP also displays a list of attributes derived from the request context and the current LWWCM user credentials, which could be used to access the back-end system.

Important: The code to retrieve the user ID and password is not supported and likely to be changed in future releases of LWWCM.

Example 7-19 test.jsp

```
<%@ page import="java.util.Enumeration" %>
<%@ page import="com.apatrix.pluto.rendering.RenderContext" %>
<%@ page import="com.presence.connect.user.User" %>
<jsp:useBean id="conn" scope="page" class="com.ibm.mm.beans.CMBCConnection" />

Connecting....<br>

<%
```



```

// -----
// Connect to CM using the connection bean
// -----
conn.setDsType("ICM");// 'Fed' to use federated connector
conn.setServerName("ICMNLSDDB");// CM database
conn.setUserid("icmadmin");// UserId to access CM
conn.setPassword("password");// Password of the UserId
conn.connect();// connect to CM

// -----
// Check if we are connected
// -----
boolean isConnected = conn.isConnected();

if (isConnected) {

    String attrName = "";
    Object attrValue = null;

    // -----
    // Loop through the request context and display it on the page
    // -----
    for (Enumeration e = request.getAttributeNames() ; e.hasMoreElements() ;) {
        attrName = (String) e.nextElement();
        attrValue = request.getAttribute(attrName);
%>
        <b><%= attrName %>:</b> <%= attrValue %><br><br>
<%
    }
}

// -----
// Disconnect from CM
// -----
conn.disconnect();

// -----
// Read the current LWWCM username and password from the render context
// -----
String username = null;
String password = null;

RenderContext rendercontext = (RenderContext)
request.getAttribute("JSP_CMPNT_RENDERERCONTEXT");

if (rendercontext != null) {
    User user = rendercontext.getPlutoSession().getSession().getUser();
    if (user != null) {
        username = user.getUserName();

```

```

        password = user.getConnectPassword();
    }
}
%>

<b>UserName = <%=username%></b><br />
<b>password = <%=password%></b><br />

```

Sample JSP code: simple search

Example 7-20 shows a JSP file that demonstrates how to execute a Content Manager query from within a JSP page. The result set is then displayed.

The query syntax we use here is the common query language of Content Manager. A basic query string looks like this:

```
/<ItemType>[@<Attr_Name>=<Value>]
```

In order to get all the insurance information (ItemType=insurance) for clients who live in San Jose (attribute=homeCity), the query string looks like this:

```
/insurance[@homeCity="\San Jose\"]
```

Note that the quotes in our query have to be escaped in order to assign them to a string variable in our JSP page.

For more advanced query strings, see Section 7.3.1 in *Content Manager Implementation and Migration Cookbook*, SG24-7051.

Example 7-20 CMDocumentList.jsp

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<%@ page import="com.ibm.mm.beans.*, java.net.URL"%>
<%
// -----
// Setup the CM Connection
// -----
String servername = "ICMNLSDb";
String servertype = CMBBaseConstant.CMB_DSTYPE_ICM;
String userid = "icmadmin";
String password = "password";
String query = "/insurances[@InsuranceType LIKE \"%xyz%\"]";

CMBConnection conn = new CMBConnection();
conn.setServerName(servername);
conn.setDsType(servertype);
conn.setUserid(userid);
conn.setPassword(password);

```

```

conn.connect();

CMBSearchResults results = null;
CMBDataManagement dm = null;
synchronized(conn)
{
    // -----
    // Get a data management object for later getting the URLs
    // -----
    dm = conn.getDataManagement();

    // -----
    // Run the query
    // -----
    CMBQueryService qs = conn.getQueryService();
    qs.setAsynchSearch(false);
    qs.setQueryString(query, CMBBaseConstant.CMB_QS_TYPE_XPATH);
    qs.runQuery();

    results = new CMBSearchResults();
    results.setConnection(conn);
    results.newResults(qs.getResults());
}
%>

<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<META name="GENERATOR" content="IBM WebSphere Studio">
<TITLE>A List of CM Documents</TITLE>
</HEAD>
<BODY>
<%
// -----
// Loop over the results
// -----
for (int i = 0; i < results.getCount(); i++)
{
    synchronized(conn)
    {
        CMBItem item = results.getItem(i);
        dm.setDataObject(item);
        URL url = dm.getContent(0).getDataURL();
%>
        - <a href="<%= url.toString() %>"><%= item.getAttrName(1) %> = <%=
item.getAttrValue(1) %></a><br>
<%
    }
} // end loop over results
%>
</BODY>

```

```

</HTML>

<%
// -----
// Tear down the CM Connection (in a real world example
// you might leave the connection in the HTTP session)
// -----
conn.disconnect();
%>

```

Figure 7-25 shows the output of the sample query page in an LWWCM Web site.

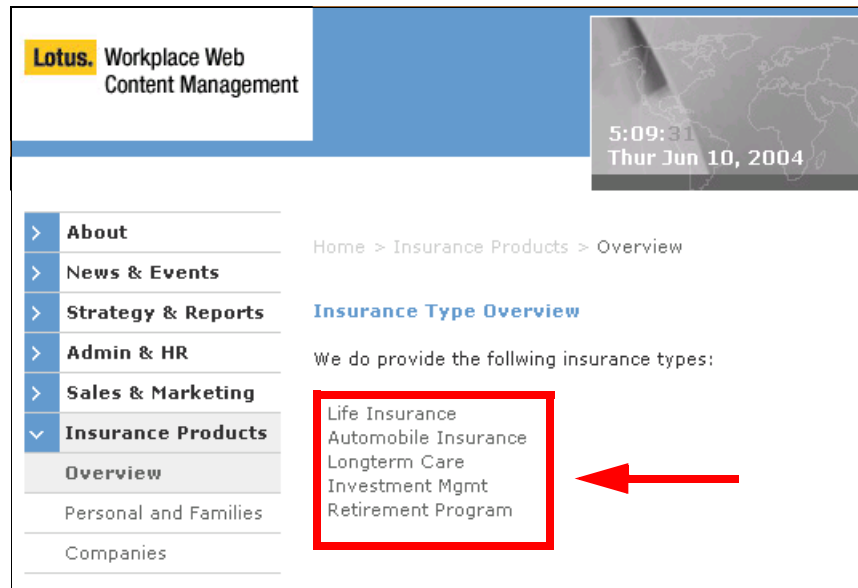


Figure 7-25 Output of CMContentList.jsp

Sample code: using wcm JSP tags in JSP pages

LWWCM JSP tags are used within JSP pages to retrieve LWWCM content and components. This can be handy if you want to implement a JSP page which is completely outside of the LWWCM Web application. This could be a JSP page which offers the user the ability to manage his or her insurance using a custom Web interface of Content Manager. To surround this interface with LWWCM components such as a navigator and menus, you could use the wcm tag library.

To use wcm JSP tags in JSP pages, follow these steps:

1. Include the wcm tag library in your JSP page. By default, this tag library can be found in the WEB-INF/tld folder of the LWWCM Web application.

```
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
```

2. Initialize the workspace. This tag expects valid credentials of a LWWCM user.

```
<wcm:initworkspace username="Administrator" password="password">  
    Could not initialize the workspace  
</wcm:initworkspace>
```

3. To tell the page about which LWWCM context to use, use the `setExplicitContext` tag. This tag can be used more than once on a page. After each occurrence of the tag, the context changes for the remainder of the page. To initially set the context to `/intranet` you would have to do the following:

```
<wcm:setExplicitContext path="/Intranet">  
    Could not set the explicit path  
</wcm:setExplicitContext>
```

4. To render an actual LWWCM component or content, you can use the `libraryComponent` tag and the `contentComponent` tag, which are similar to the `AprixLibCmpnt` tag and the `AprixCmpnt` tag used when creating the HTML of a page design within the LWWCM environment. Assuming you wanted to show the same navigator on the same page twice in different contexts in addition to a body field, you would have to do the following:

```
<wcm:setExplicitContext path="/Intranet">  
    Could not set the explicit path  
</wcm:setExplicitContext>  
  
<wcm:libraryComponent name="NAV - Left Hand">  
    Could not find LH Nav component  
</wcm:libraryComponent>  
  
<wcm:setExplicitContext path="/Intranet/Home/About">  
    Could not set the explicit path  
</wcm:setExplicitContext>  
  
<wcm:libraryComponent name="NAV - Left Hand">  
    Could not find LH Nav component  
</wcm:libraryComponent><br>  
  
<wcm:contentComponent key="Body" type="content">  
    Content not found  
</wcm:contentComponent>
```

The result should look similar to Figure 7-26.:

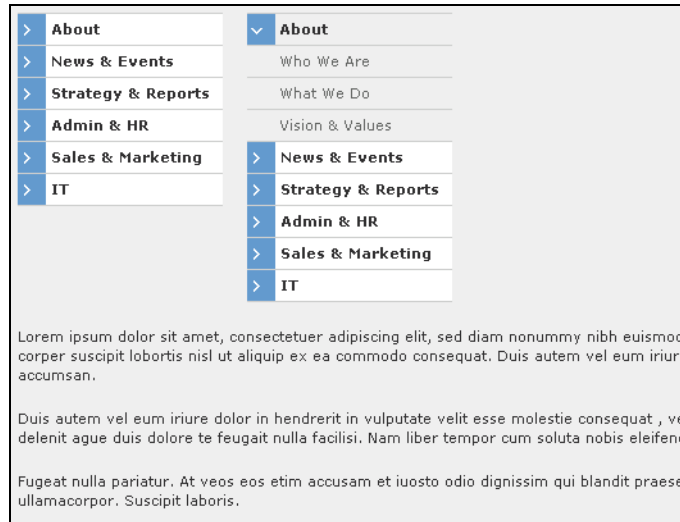


Figure 7-26 WCM tag sample page

For more detailed information regarding wcm JSP tags, refer to the LWWCM online Information Center.

7.5.5 Putting it all together

In this section, we describe the necessary actions to make the new JSP component available on your Web page. We work with our insurance case study for demonstration purposes. For our case study, we want to get a list of all available insurance policies on a sub-page called *insurances*. The query string in CMContentList.jsp has to be adjusted to the following:

```
String query = "/insurances";
```

Within Content Manager System Administration client, implement the following setup if it is not already done:

1. Create sample attributes, IT_Number, IT_Name, IT_Description, and IT_TargetGroup.
2. Create an item type, InsuranceType, with the above attributes (see Figure 7-27 on page 163).
3. Import some entries for this InsuranceType.

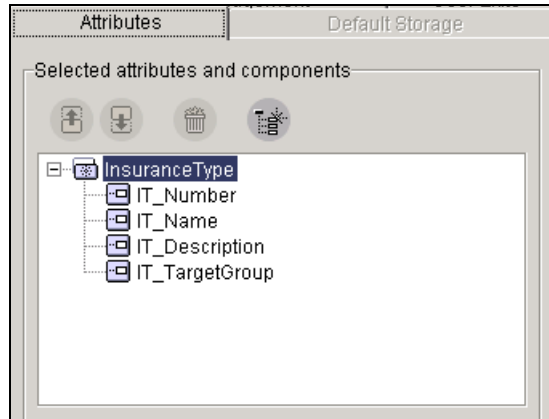


Figure 7-27 Item Type and Attributes

Within LWWCM, do the following:

1. Create the Simple Search JSP page (CMContentList.jsp) as described in "Sample JSP code: simple search" on page 158 and save it as [web-app]\jsp\CMContentList.jsp.
2. Create the JSP component as described in 7.5.3, "Setting up a new JSP component" on page 152. Name it MyJSPCmpnt. For the path field, enter:
/jsp/CMContentList.jsp
3. Create a new content template. Content created with this template serves as a placeholder for the initial content of the new site area. Here, we will not reference the JSP component through the content context but directly through a tag in the page design; therefore, we do not have to add the component to the content.
4. Create a new page design called CM Content Layout (making a copy from the Menu Layout and editing from there will speed up the process) and reference your JSP component in the HTML body. Replace the following line:
`<ApatrixCmpnt context="current" key="Menu" type="content"/>`
with
`<ApatrixLibCmpnt name="MyJSPCmpnt"/>`
5. Create a new site area under which you want the results of your JSP component to be displayed. In the properties tab of the site area, specify the newly created content and page design (see Figure 7-28 on page 164).

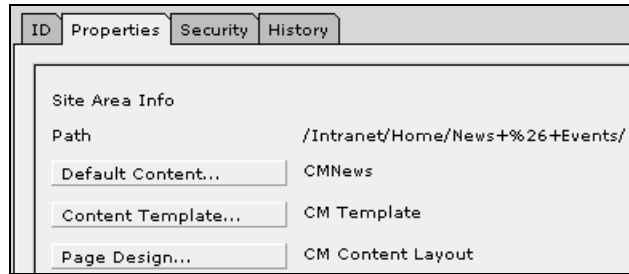


Figure 7-28 Setting up new site area

6. Create new content called CMNews using the content CM Template and put it into the site area you just created. Since the content was not available when you created the site area, you must go back to the site area and add CMNews as the Default Content.
7. Click **Preview** on either the new content or the new site area. The resulting Web page should look similar to Figure 7-25 on page 160.

7.6 Troubleshooting

This section offers some hints and tips on how to resolve some of the problems you may encounter.

7.6.1 Log files and trace levels

The following log files and trace level settings are useful to isolate problems:

► WebSphere Application Server

Use WebSphere Application Server log files. Since we are executing a JSP in the context of the LWWCM Web application, which is running on a WebSphere Application Server, this log should be consulted first of all.

► LWWCM

Log files are located in the following directory: [LWWCM_HOME]\connect\log

- connect.log
- error.log
- debug.log

The trace level of the files mentioned above can be adjusted separately in the Log manager configuration section of the following file:

[LWWCM_HOME]\config\connect.cfg

7.6.2 Basic connection testing

To test if the connection from your LWWCM server to the Content Manager server is working, we recommend executing the TConnect sample application from the LWWCM server. To test a connection, you have to provide TConnect with a couple of parameters. It will then try to access the target Content Manager, open a connection, and close it.

TConnect comes with the installation of Information Integrator for Content. The syntax of TConnect is as follows:

```
java TConnect <dstype> <server> <userid> <password>
```

Example:

```
java TConnect ICM icmnlsdb icmadmin password
```

If you installed Content Manager and LWWCM on separate machines, it is necessary to catalog the Content Manager database on your LWWCM server.

To get a list of all databases cataloged, enter the following at the command prompt:

```
db2 list database directory
```

The Content Manager database must be part of the resulting list.

7.6.3 Executing the JSP program outside of LWWCM

When working with JSP component in LWWCM, make sure the referenced JSP file works outside of LWWCM. Try to call the JSP directly using your Internet browser:

```
http://<server>/<context>/connect/<pathToJsp>/<jsp>.jsp
```

If it does not work, you need to debug it and make sure it works before having the JSP component refer to it within LWWCM.



System architecture considerations

In the previous chapters, we addressed the basics of how to use Content Manager as a repository for LWWCM and how to publish Content Manager content to LWWCM Web sites. In this chapter, we discuss the system architecture considerations involved when integrating LWWCM with Content Manager.

Specifically, in this chapter, we cover the following topics:

- ▶ LWWCM and Content Manager on the same machine
- ▶ LWWCM and CM on different machines
- ▶ Integration scenarios
- ▶ Mixed operating systems (Windows and AIX)

8.1 Introduction

When integrating Lotus Web Content Management (LWWCM) with Content Manager, there are many ways to set up the integrated solution. We can install LWWCM and Content Manager on one machine, install them on separate machines, have one or more LWWCM servers with one Content Manager system, or have one or several Resource Managers within the Content Manager system to integrate with LWWCM server(s). In addition, we can have servers and systems in a mixed platform environment to satisfy business requirements.

To achieve proper communication between LWWCM and Content Manager, we need to ensure the following components are installed:

► *DB2 Runtime Client*

The Content Manager database alias always needs to be cataloged; therefore, at the minimum, the DB2 runtime client must be installed.

► *Information Integrator for Content Development Workstation or Server*

Using Content Manager as the repository for LWWCM objects only has different requirements regarding the installed connectors and Information Integrator for Content than if using it as a repository for external resources such as videos, pictures, and documents.

Figure 8-1 shows the components necessary for the integration.

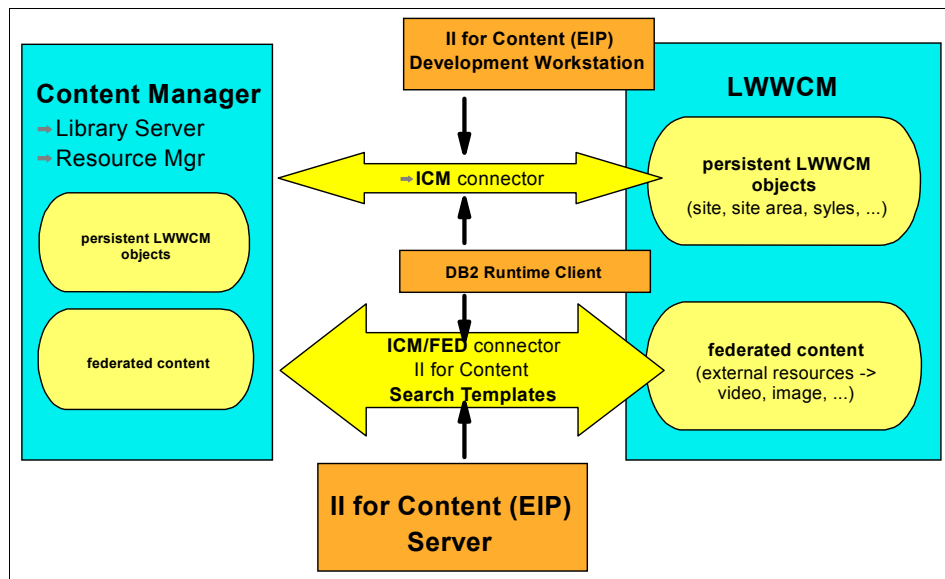


Figure 8-1 Necessary component for one and many boxes

8.1.1 Using Content Manager as the repository for LWWCM objects

When using Content Manager as the repository, the hosting LWWCM must have at least the *ICM connector* installed. The ICM connector can be selected and installed during the installation of Information Integrator for Content Development Workstation. Refer to 7.2, “Prerequisites” on page 117 for more details on the installation procedure.

The classpath of the application server which is running LWWCM has to be adjusted to also include the necessary classes to communicate with Content Manager.

8.1.2 Using Content Manager as the repository for external resources

In order to use Content Manager as the repository for external resources such as videos, images or documents, you might want to use the federated content component within LWWCM, which is discussed in 7.3, “Federated content component” on page 123. This component makes use of the Search Templates provided by an Information Integrator for Content server through the federated connector (*FED*). So, in order to get this functionality, you would also have to install the Information Integrator for Content server on this machine and the *FED* connector of Information Integrator for Content in addition to the *ICM* connector.

The classpath of the application server which is running LWWCM has to be adjusted to also include the necessary classes to communicate with Content Manager.

8.1.3 Server operating system considerations

In our test environment, everything was also tested on AIX. The operating system has no influence on the one or many box decision, since all the products and components are available on AIX as well.

8.2 LWWCM and Content Manager on the same machine

For testing purposes or small environments, it might be sufficient to install LWWCM and Content Manager on one box (see Figure 8-2 on page 170). This is a supported architecture. In this section, we discuss what is necessary if you want to use this approach.

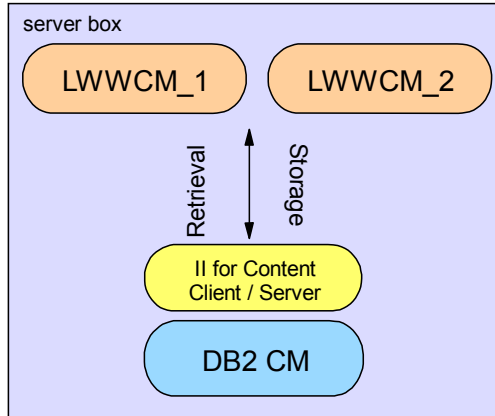


Figure 8-2 LWWCM and Content Manager on the same box

8.2.1 LWWCM considerations

Multiple LWWCM systems can be installed on the same machine (although this is not recommended except for testing purposes). They are completely independent of each other and you can handle them as if they were installed on different boxes. There are two ways to install multiple LWWCM systems under the same WebSphere Application Server:

1. Create a dedicated application server for each LWWCM server. This has the advantage that the servers are running in separate JVM environments. LWWCM servers can be stopped and started independently. The servers have to be accessed on different ports but can use the same context information. The two URLs below only differ in port number. Each of the URLs would access a dedicated LWWCM server, one running on port 9080, the other on port 9081:

```
http://demo1:9080/ilwwcm/connect/intranet
http://demo1:9081/ilwwcm/connect/intranet
```

2. Run each LWWCM server in its own context, but under one WebSphere Application Server. In most testing scenarios, the LWWCM server is installed under server1. Using this approach, you can have many LWWCM servers running under server1, but using different contexts with the same port. The following two URLs only differ in their context information (the context of the first server is ilwwcm_1, the context of the second server is ilwwcm_2), which is defined during deployment of the WAR file using the same port number (9080) and installed on the same WebSphere Application Server:

```
http://demo1:9080/ilwwcm_1/connect/intranet
http://demo1:9080/ilwwcm_2/connect/intranet
```

The two servers are completely independent and would therefore also support syndication for one another.

8.2.2 Content Manager considerations

Multiple Content Manager systems can be installed on the same machine as an LWWCM server or multiple LWWCM servers. We do not recommend doing this due to the extensive memory usage of Content Manager. If you want to test the multiple Content Manager scenarios discussed later in this chapter, we advise you to have the additional Resource Managers and/or Library Servers installed on dedicated machines.

8.2.3 Security considerations

If we publish our content to the Internet, it will be accessible from the Internet. This will result in the server not being in a secure zone; there is no guarantee that the data in it is secure.

With only one server for all the information storage and publication, we are more exposed to intruders from outside the company. At most, we could only install a firewall between our server (intranet) and the outside world (Internet).

8.2.4 Maintenance

Having the entire Web site on only one server instead of one for each Web site process stage could be hard to maintain. The reason is that most companies will most likely use more than one process stage for Web sites. For example, a development environment and a production environment are common for most organizations for the content publication process. But since we have only one LWWCM server in this approach, that would not be possible.

This approach forces us to have all the content publication process in the same LWWCM server. That is also dangerous because we have the authorized published data content mixed with the draft data in the same LWWCM server. In 8.3, “LWWCM and CM on different machines” on page 172, we discuss the usage of different machines.

8.2.5 Advantages and disadvantages

The advantage of this all-in-one approach is that it is easy to set up. You can quickly set up the system and test to see how everything works together. This approach is recommended for small and simple Web sites that do not need to have several stages for the Web site publishing process.

The disadvantage of this all-in-one approach is that there is a high risk of losing all the data if the machine crashes. If this happens and we do not have a back-up, all the information will be lost, as well as the products configuration. As discussed earlier regarding security, there maybe security issues with this simple setup. In addition, there maybe maintenance issues as well. We do not recommend this approach for an enterprise system.

8.3 LWWCM and CM on different machines

For performance and security reasons in production environments systems, we recommend you have LWWCM and Content Manager installed on different boxes (see Figure 8-3). In this section, we discuss what is necessary if you decide to use this approach.

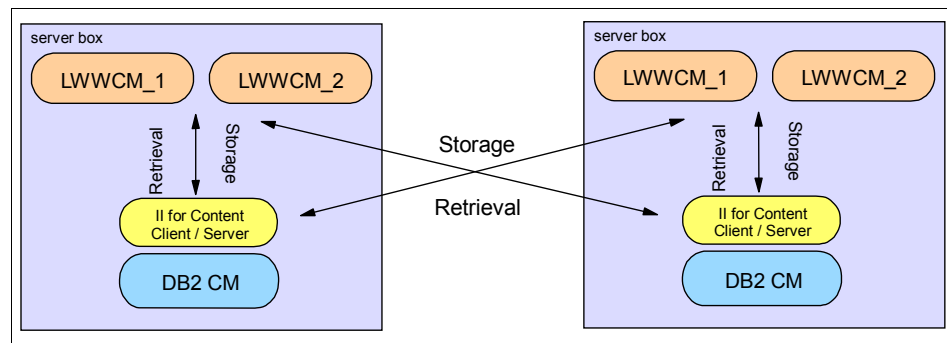


Figure 8-3 LWWCM and Content Manager on different boxes

8.3.1 LWWCM considerations

If you want to install several LWWCM servers on the same machine, refer to 8.2, “LWWCM and Content Manager on the same machine” on page 169. The servers would differ in either the port number or the application server context in which they are running.

Having each LWWCM server running on its own machine requires no special attention regarding the context and port numbers since they are running on different application servers. Each server would be accessible via a different host name. The following two URLs would access two different LWWCM servers installed on two different machines (demo1 and demo2) using the same context and port numbers:

```
http://demo1:9080/ilwwcm/connect/intranet
http://demo2:9080/ilwwcm/connect/intranet
```


Syndication between the two LWWCM servers is only supported if they are of the same version. In our scenarios, we use LWWCM V2.0 for all LWWCM servers.

8.3.2 Content Manager considerations

There are no special considerations regarding Content Manager. The installation and configuration tasks of a Content Manager installed on a different machine other than LWWCM do not differ from those of a Content Manager installation on the same machine as the LWWCM server.

8.3.3 Security considerations

If we have more than one machine, you can separate the Content Manager server from the LWWCM server(s).

If LWWCM is managing content for an Internet site, it is highly recommended that the external Web server be on a DMZ to avoid external attacks. This way, we ensure the data stored in the Content Manager repository is behind a firewall, and make it unavailable to intruders. See the setup in Figure 8-4.

DMZ is an abbreviation for Demilitarized Zone. It is a concept that represents a host/network between your router and the firewall or a host/network outside the firewall. The machines in DMZ are exposed to an untrustworthy outside world (Internet), which is considered a high-risk zone for attacks. No confidential information should reside on any server in a DMZ in an unprotected form. Only approved ports should be opened between the first (Internet) and second (Internal network) tier.

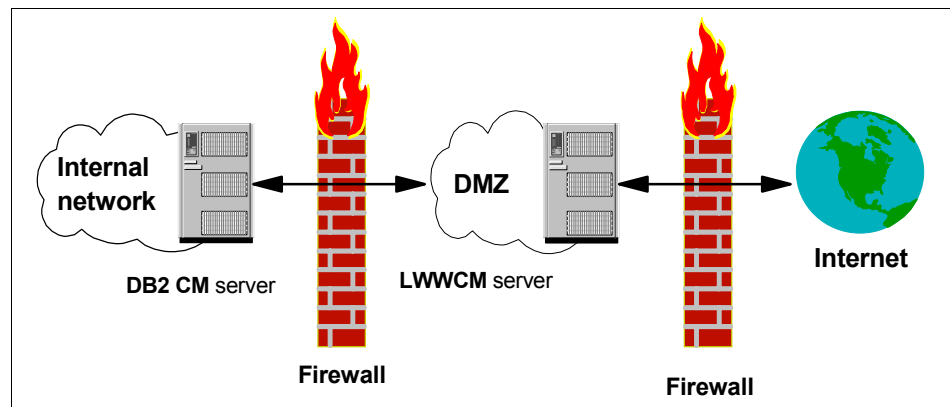


Figure 8-4 Secure architecture example: LWWCM/CM in separate boxes.

8.3.4 Maintenance

Web sites on different environments can be managed more effectively. Also, the general performance can be improved by having separate environments which allow a granular approach to performance considerations. As a way to make the maintenance easier, the first decision should be to analyze the different environments we would need depending on various factors such as the Web site size, distribution, and needs.

8.3.5 Advantages and disadvantages

The most important advantage of this approach is that we have the data distributed and we can also ensure a good level of security. Another important advantage is the possibility of dividing the Web site publication process into different types of environments, according to your company's needs.

8.4 Integration scenarios

In this section, we discuss the different possible scenarios for the integration of LWWCM and Content Manager. Other combinations, not discussed here, are possible, depending on your business requirements.

The decision of which integration scenario to choose depends on factors such as the geography of the company, LWWCM processing stages, required independent development, testing, and production environment, the number of physical servers you can allocate, and what level integration will be needed.

8.4.1 One LWWCM server

The first approach is the most simple one since it implies that all products are installed. In order to make the integration possible, we must also install the Information Integrator for Content server on the same machine. This installation provides the necessary connectors to achieve the communication between Content Manager and LWWCM.

Both products could be installed on the same machine or on different machines. Using different machines, it will be necessary to install DB2 runtime client and catalog the Library Server database. For more information about this configuration, see 6.2, "Accessing a remote Library Server database" on page 97 and 8.2.1, "LWWCM considerations" on page 170.

To achieve the integration, the following process is executed:

1. A request to LWWCM Web application is made through a Web browser.

2. The LWWCM Web application, which is hosted in an application server, uses the Content Manager Java API (the Federated connectors, JAR files in the classpath) to access the Library Server using the *same* user ID and password as defined in the aptixje.properties file.
3. Each object is stored in Content Manager AJPEData item type which has its own attributes such as name, type, and description. This is the defined default item type name in the aptixje.properties file. This will be described in more detail in 6.4, “How LWWCM objects are stored in CM” on page 108.
4. Binary files and image objects are also stored as an AJPEResources item in Content Manager. The actual objects are stored physically in Resource Manager.
5. The attribute JPE Server ID in both item types identifies the LWWCM server.

Figure 8-5 shows the details of the integration.

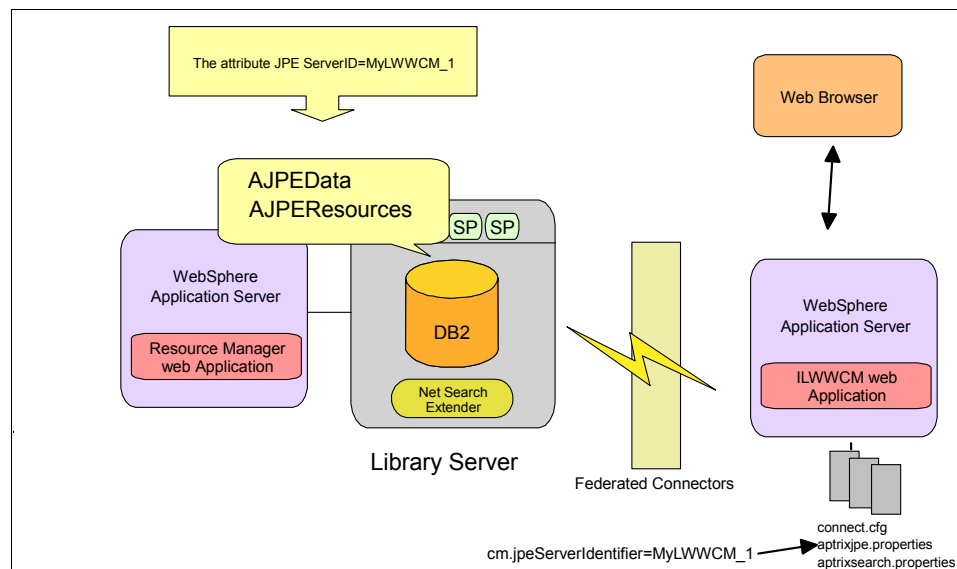


Figure 8-5 Integration details

For this configuration, with one LWWCM server, we could have either one Library Server and one Resource Manager, or one Library Server and multiple Resource Managers.

One Library Server and one Resource Manager

In this scenario, one LWWCM server is integrated with one Content Manager system which has one Library Server and only one Resource Manager. With this scenario, we have the further option of having everything installed on one

machine or have LWWCM and Content Manager on different machines. The considerations for each scenario (one box or more than one) are described in the previous sections. The data in Content Manager is all stored on one unique machine.

One Library Server and multiple Resource Managers

In this scenario, one LWWCM server is integrated with one Content Manager system that has one Library Server and multiple Resource Managers. With this scenario, we need to have multiple machines: an additional one for each additional Resource Manager. One reason to set up the system this way is to have the AJPEResource item types stored in another machine. This approach is useful if you host geographically diverse Web sites that require access to large image or video objects and you want to set up Resource Managers that are closer to where users are most likely going to retrieve the files.

To set up this configuration, the Resource Manager associated with the actual AJPEResource item type needs to be replaced with the extra Resource Manager:

1. Open the Content Manager System Administration client.
2. Select **Data Modeling -> Item Types**.
3. From the right panel, right-click **AJPEResource** and select **Properties** from the context menu.
4. Select the **Document Management** tab.
5. Select the existing Part type (**ICMBASE** by default) and click **Edit**.
6. In the pop-up window, you can change the default Resource Manager (and the corresponding collection) and select another one which is installed on another machine.
7. Click **Apply** and then **OK**.

In Figure 8-6 on page 177, we show an example of a Content Manager server which has two associated Resource Managers, RMDB and RM2.

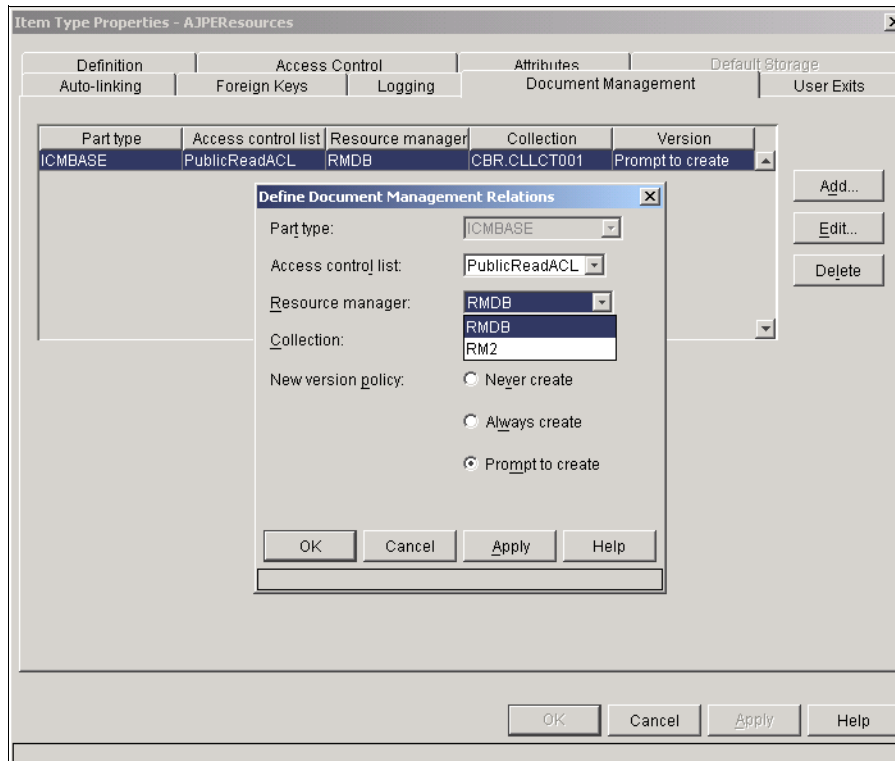


Figure 8-6 Configuration of the AJPEResources to point to another RM.

8.4.2 Several LWWCM servers

We recommend setting up several LWWCM servers if the Web sites are expected to be large and/or complex. When dealing with such Web sites, it is generally necessary to divide the processes into different logical environments for each specific requirement. For most of the Web sites, it is common to find at least two environments, one for development and authoring, and another one for production, the live Web site.

In this LWWCM Java edition, Web sites can share information between the environments through the syndication activity. Syndication allows data to be replicated among different LWWCM servers. The syndication is based on the Information and Content Exchange (ICE) standard and uses HTTP. For more detailed information about syndication, refer to 13.1, “Syndication overview” on page 272

Advantages

The syndication in this approach is very useful because it allows us to have many distributed servers with the same data replicated in all of them. We have a choice of syndication or not, depending on the business needs or stages.

There are different types of possible environments:

- ▶ **Authoring:** In many large companies, it could be necessary to have multiple servers that spread across multiple physical locations. Content can be created and maintained locally and then syndicated to a central authoring server.
- ▶ **Staging:** Provides a chance to review changes in the context of the entire site before they are sent into the live environment. We highly recommend setting up this environment in the case where large changes are expected to be made on regular intervals.
- ▶ **Live:** Provides end consumers with the live Web site. This is tuned for efficiently rendering Web pages. The environment often include other performance enhancing software and hardware to help with caching, load balancing, and failover.
- ▶ **Development:** Provides for development and testing. After the appropriate testing, syndication can be used to send the data to the live environment or optionally a staging environment.
- ▶ **Testing:** Useful in cases where a development environment needs to be separated from a testing environment.

Having several LWWCM servers using a unique Content Manager repository does not apply only to the Web sites which need to have different stages. It is also useful, for example, if we want to have different Web sites but want to store information in the same repository. In this case, the syndication would not be necessary because the data in each Web site (LWWCM server) would be different.

Disadvantages

One disadvantage of having multiple LWWCM servers, with different environments syndicating the data, is that if we use *external files* within the content site, we need to deliver these external files to all the servers manually and store them in each server's file system. For example, if we use a JSP file (external to LWWCM) as we show in 7.5, "JSP component" on page 151, the JSP file needs to be copied to all the servers that are being syndicated. Otherwise, the JSP component will try to find the JSP file unsuccessfully and will not be able to render the JSP content. Another similar case is the use of an HTML file (external to LWWCM) to show the search results using connect tags as we will see in 7.4.3, "Putting it all together" on page 148. In this case, the HTML file would also need to be copied to all the servers every time it is modified *manually*.

Environment samples

A simple example of having two different environments could be to have the content creation and maintenance carried out in the authoring environment and this content syndicated to the live environment.

In order to include a staging environment, we could add it between the authoring and the live environments.

Within a large company, there may also be a need to be multiple servers spread across many physical locations. Content can be created and maintained locally and then this content can be syndicated to a central authoring server.

Figure 8-7 shows the scenario where there are three environments, Authoring, Staging, and Live, and each is set up on a separate machine.

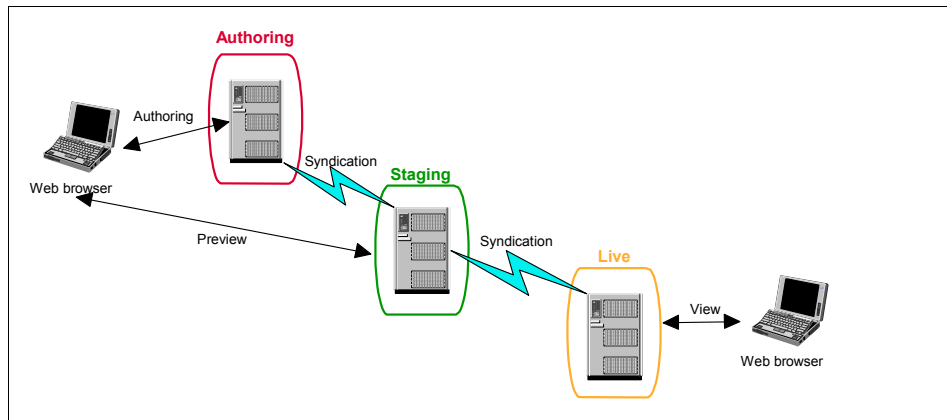


Figure 8-7 Multi-environment, multi-machines setup

One Library Server and one Resource Manager

This approach implies having many LWWCM servers pointing to a unique Content Manager repository, with one Library Server and one Resource Manager.

To achieve this integration, the Information Integrator for Content Development Workstation must be installed on each LWWCM machine to provide the necessary connectors for the communication. Figure 8-8 on page 180 illustrates this scenario setup.

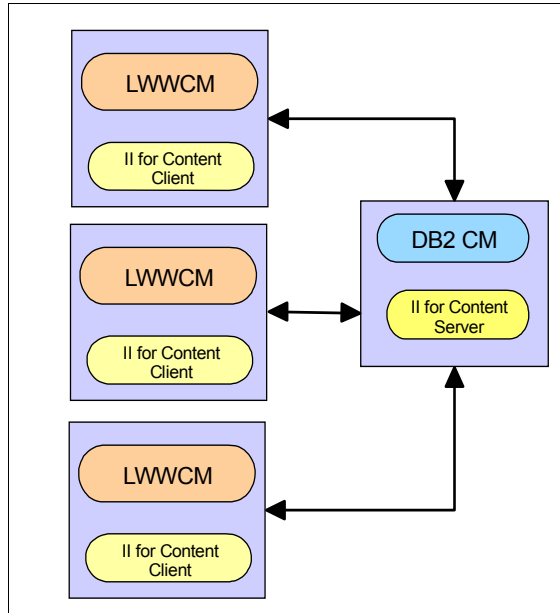


Figure 8-8 Multiple LWWCM servers/machines scenario architecture

When several LWWCM are installed and configured to use the same Content Manager repository, each LWWCM must have a different name. This is defined in the `aprixjpe.properties` file on each LWWCM server:

```
cm.jpeServerIdentifier=MyLWWCM_1
cm.jpeServerIdentifier=MyLWWCM_2
```

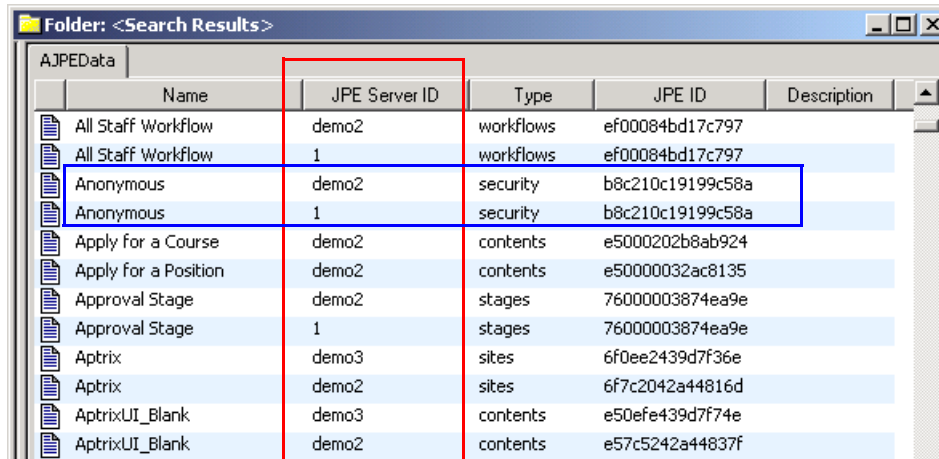
In this scenario, *MyLWWCM_1* and *MyLWWCM_2* are the names of two LWWCM servers. Any combination of numbers and letters can be used to name the different servers.

Tip: We recommend using the host name of the server as your `jpeServerIdentifier`. If you have multiple LWWCMs on one machine, append a number to the host name. This is because the host name is unique and can easily be associated with a physical machine.

From Content Manager, the attributes JPE ServerID in `AJPEData` and `AJPEResources` identify the object in the proper LWWCM Server. If you select **Start -> Programs -> IBM Content Manager V8 -> Client for Windows**, log on to the Library Server, and run a search based on item type `AJPEData`, you will get all items stored in Content Manager with this item type. The results of this search are shown in Figure 8-9 on page 181. Note, for example, that there are two items called *Anonymous*. Both items have a different JPE Server ID. This

means that even if the items are called the same, they belong to different LWWCM servers. The first one belongs to the server identified as *demo2* and the second one to server identified as *1*.

Note: The only attribute which has different values in this example is JPE Server ID. This is because both LWWCM servers are syndicated and so the data is the same in both.



Name	JPE Server ID	Type	JPE ID	Description
All Staff Workflow	demo2	workflows	ef00084bd17c797	
All Staff Workflow	1	workflows	ef00084bd17c797	
Anonymous	demo2	security	b8c210c19199c58a	
Anonymous	1	security	b8c210c19199c58a	
Apply for a Course	demo2	contents	e5000202b8ab924	
Apply for a Position	demo2	contents	e50000032ac8135	
Approval Stage	demo2	stages	76000003874ea9e	
Approval Stage	1	stages	76000003874ea9e	
Apatrix	demo3	sites	6f0ee2439d7f36e	
Apatrix	demo2	sites	6f7c2042a44816d	
ApatrixUI_Blank	demo3	contents	e50efe439d7f74e	
ApatrixUI_Blank	demo2	contents	e57c5242a44837f	

Figure 8-9 Different LWWCM using the same Content Manager

Figure 8-10 on page 182 illustrates the scenario where several LWWCM servers use the same Content Manager.

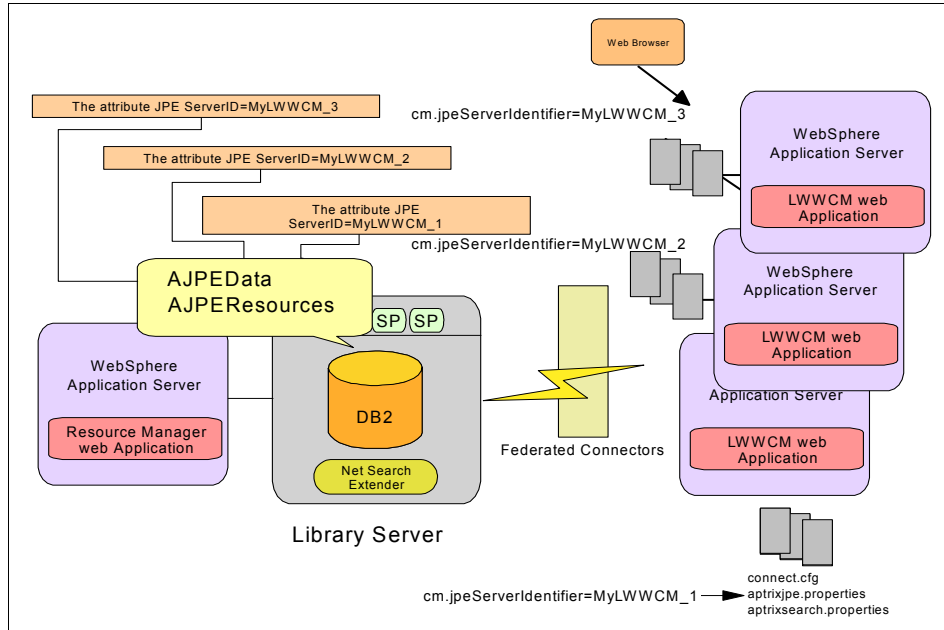


Figure 8-10 Several LWWCM servers using the same Content Manager

Note that in the above scenario, each LWWCM has one server name defined in the proper `aptrixje.properties` file. This name is used in **AJPEData** and **AJPEResources** item types from Content Manager to identify which server each object belongs to.

One Library Server and multiple Resource Managers

This approach covers the cases where you want to share the load of many LWWCM servers among many Resource Managers, even if the LWWCM servers all use the same Library Server. This might make sense in the following occasions:

- ▶ For load balancing and performance reasons, the design and content of the LWWCM servers need to be synchronized using LWWCM's syndication feature. You might also want to share the load of the Resource Managers, having each LWWCM server accessing its own Resource Manager.
- ▶ Two or more LWWCM servers host different Web sites. For back-up, recovery, performance, and load balancing reasons, they need to use independent Resource Managers through the same Library Server.
- ▶ One Resource Manager is used to store LWWCM objects such as LWWCM site, site areas, content templates, workflows, and workflow stages. The other

Resource Manager is only to serve as the source for external content such as videos and images.

Having multiple LWWCM servers using the same Library Server but different Resource Managers for the storage of LWWCM objects can be accomplished through a special configuration on each LWWCM server. By default, the LWWCM server creates and uses two item types in the Content Manager server called *AJPEData* and *AJPEResource* to store its objects. Among other settings, these names can be adapted to match a specific item type name. You can take advantage of this fact, and define a different Resource Managers for different item types.

To change the name of the item type which will be created and used by the LWWCM server, open the `aptrixjpe.properties` file in `<LWWCM_HOME>\config` directory and edit or add lines as shown in Example 8-1 for the first LWWCM server.

Example 8-1 aptrixjpe.properties of the first LWWCM server

```
..
cm.server=ICMNLSDDB
...
cm.itemType=demo1_data
cm.resources.itemType=demo1_resources
...
```

Configure an additional LWWCM server using the lines shown in Example 8-2.

Example 8-2 aptrixjpe.properties of the second LWWCM server

```
...
cm.server=ICMNLSDDB
...
cm.itemType=demo2_data
cm.resources.itemType=demo2_resources
...
```

Figure 8-11 on page 184 outlines this example configuration. For the first LWWCM server, LWWCM 1, it uses item type `demo1_data` and resource item `demo1_resources`. For the second LWWCM server, LWWCM 2, it uses `demo2_data` for the item type and `demo2_resources` for the resource item. Both LWWCM servers point to the same Content Manager Library Server, ICMNLSDB. Within the shared Library Server, `demo1_data` of item type and `demo1_resources` of resource item type belong to the Resource Manager, RM1. The `demo2_data` and `demo2_resources` belong to the Resource Manager, RM2. This means that each LWWCM server will have its own Resource Manager to store video and files.

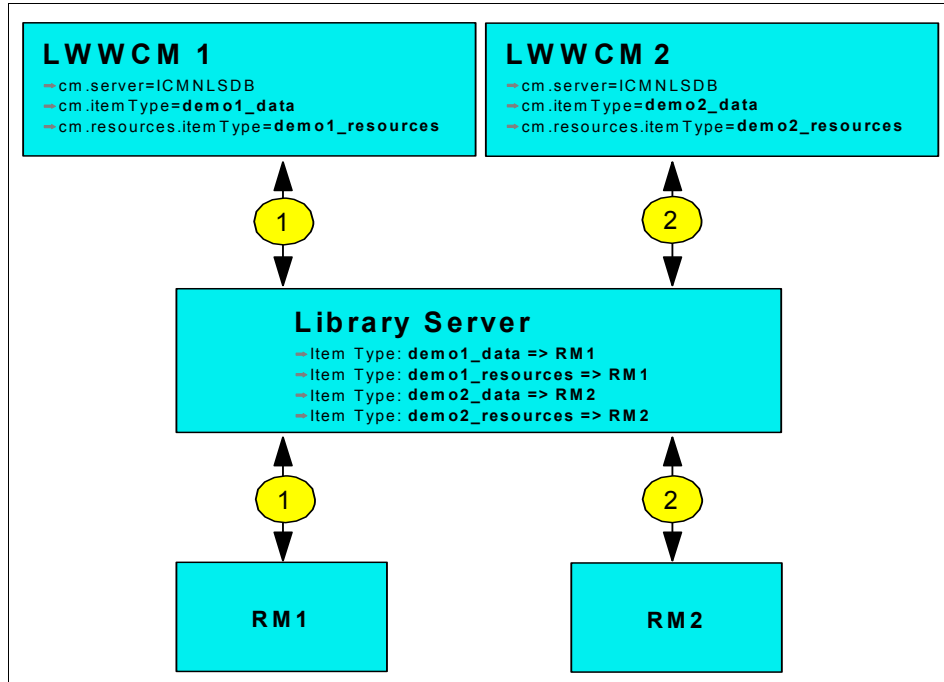


Figure 8-11 Multi-LWWCM using multi-Resource Managers

For more information regarding specific Content Manager related configuration, refer to the chapter *Storing Data in IBM Content Manager* from the LWWCM online Information Center.

Multiple Content Manager servers

This approach covers the cases in which companies have many offices geographically distributed in different locations. They may also have multiple Content Manager and LWWCM servers, one for each location. In this case, each LWWCM server uses a Content Manager server as repository and the common data (the Web site content and design objects) is syndicated between the different LWWCM servers. This way, we ensure that the same Web site is stored in each location but has the same data everywhere. From the security point of view, this is a good approach because each location is independent of the others regarding data storage.

The configuration needed for this approach is the same as for one LWWCM and one Content Manager server since each local configuration is independent of the others. The only thing in common, in this case, is the usage of syndication to keep the same updated data in every server. To achieve the syndication, one of

the servers need to be the *Syndicator* and the other server the *Subscribers*. Figure 8-12 illustrates this scenario.

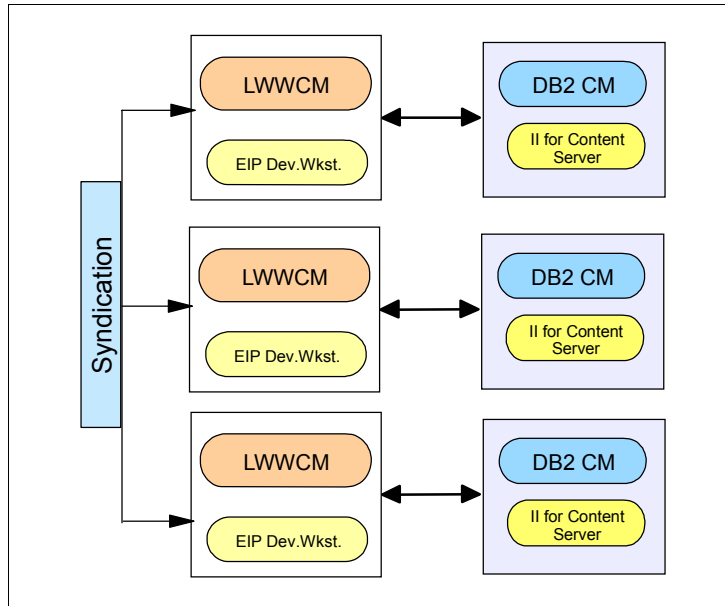


Figure 8-12 Multi-LWWCM to multi-Content Manager via syndication

8.4.3 Using Content Manager as a repository for external resources

You can have multiple Content Manager servers with data populated or imported using Content Manager as the front end application, and publishing their content to LWWCM Web sites. This approach is useful in cases where we need to use, for example, video files or images that are archived in different Content Manager servers to LWWCM Web sites. In this case, we are not only using Content Manager as a repository, we also use it as the source for external data. The different ways to retrieve the Content Manager data stored in several Content Manager servers are explained in Chapter 7, “Publishing Content Manager content” on page 115. For information on integrating Content Manager videos to LWWCM Web site, refer to Chapter 12, “VideoCharger integration” on page 255. Note that, in order to get this functionality, you must install the Information Integrator for Content server on the LWWCM server(s).

8.5 Mixed operating systems (Windows and AIX)

It is possible to have all the previously discussed environments and combinations of LWWCM and Content Manager server setups configured on

different operating platforms. Once LWWCM and/or Content Manager is/are installed on AIX and/or AIX, the usage of Content Manager and LWWCM is the same as if everything were set up on a Windows platform. It is transparent to end users as to where the servers are installed and run. Generally, we recommend that you use the same platforms for all servers. However, if the existing LWWCM and Content Manager servers are already using different platforms, you should still be able to perform the integration without any problems.



Portal integration

This chapter describes the features of the LWWCM content portlets and Content Manager portlets in detail. We also talk about user management from a portal perspective and the impact on Single Sign On with Content Manager and LWWCM (see also Chapter 11, “Security” on page 237).

9.1 Architecture

The following figure illustrates the actual global high-level architecture including WebSphere Portal Server (WPS).

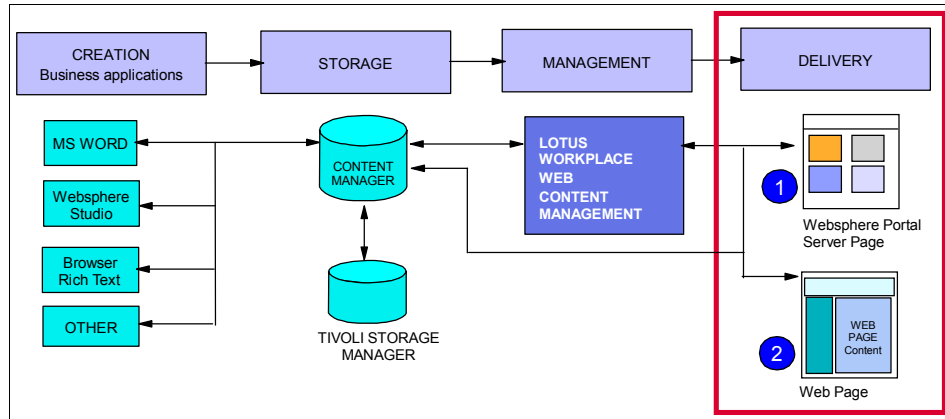


Figure 9-1 WPS/LWWCM / Content Manager architecture

9.1.1 Content creation

In the current version of LWWCM, in conjunction with the Content Viewer portlets, we only have the possibility to view content within the portal. Creation, editing and deleting of content as well as page design (creating page design, content templates, site framework) are still done in the LWWCM Administration client. In future releases of LWWCM, the LWWCM Administration client will move from an applet-based application to a portlet application which can only be used within WebSphere Portal. This chapter focuses on integrating the content viewing into WebSphere Portal V5.

9.1.2 Content storage

As discussed in Chapter 1, “Lotus Workplace Web Content Management overview” on page 3 and Chapter 8, “System architecture considerations” on page 167, content can be stored in either the file system, a relational database system or IBM Content Manager. From the portal point of view, this is transparent. It does not make a difference which repository type you are using.

9.1.3 Content management

The site framework, the relationship between content and associated site areas as well as content and associated categories are implemented in the LWWCM

administration client. It is independent of WebSphere Portal Server when WebSphere Portal Server is used as the user interface.

9.1.4 Content delivery

There are basically two ways to present LWWCM content to the user (see the box in Figure 9-1 on page 188):

1. Use WebSphere Portal Server as the rendering engine for content stored in LWWCM.
2. Use LWWCM itself as the rendering engine for its own content.

Both approaches can be used in parallel, that is, LWWCM serving content to both the portal and a “traditional” Web page concurrently. The advantages of having content and navigation rendered into the portal are discussed in 9.2, “Why integrate LWWCM in WebSphere Portal Server?” on page 190.

If you decide to publish Content Manager content on your portal page, which is independent from LWWCM, you have two options, which we discuss in the following chapters:

1. Access Content Manager content through LWWCM using the content viewer portlets.
2. Access Content Manager content directly using Content Manager portlets.

Access external Content Manager content through LWWCM

This is the preferred approach, because it centralizes all access to the content in LWWCM. It lowers the risk of having inconsistent Web pages because it channels content and components published in WebSphere Portal Server through the security, layout and navigational mechanisms of LWWCM.

► Security

The user credentials used to log into the portal can be passed through to the LWWCM system. Each object in LWWCM has its own security. In consequence, this also applies to Content Manager content which is referenced through LWWCM using either the federated component, the connect tags or the JSP page approach as discussed in 7.5, “JSP component” on page 151.

► Look and feel/corporate identity

Since the content viewer portlet also uses a page design which is administered in the LWWCM administration interface, the responsibility for the look and feel of the page layout remains in the hands of the LWWCM page designer.

- Navigation

As you will see in “Navigation portlet” on page 194, it is possible to have a navigation portlet which reflects the site framework set up in LWWCM. Your connection to Content Manager content is done using components which are finally referenced through a content page which, in turn, is attached to one or more sites or site areas in the site framework. The result is that the location of the Content Manager content is defined only once when setting up the site framework of your site. The location which is set in LWWCM will then automatically imply where the content will appear on the portal page.

9.2 Why integrate LWWCM in WebSphere Portal Server?

This chapter explores why you would choose WebSphere Portal Server as the front end for LWWCM content from the Web user’s perspective.

- Single Sign On

Once a portal user is logged into the portal, the credentials can be passed through to LWWCM to prevent users from having to enter their credentials a second time. At the same time, the portal credentials can be used to authenticate against other portal applications. The goal is that the user has to log in only once for all applications available through the portal user interface.

Note: Refer to 9.4.2, “User management and security” on page 200 for more information.

- Flexible presentation and layout model
- High-end WCM functionality
- LDAP
- Content Manager integration
- Free choice of presentation layer
- Portlet
- Traditional Web site
- WAP and others
- Intuitive click to action (C2A) navigation
- Easy creation of interconnected portlets
- Allows content and components to be delivered through WebSphere Portal Server

9.3 Levels of integration

There are basically four levels of integration which match the approach of portlet patterns for integration approaches of external applications in general.

- ▶ **Link**
Showing a link on the portal page only. Clicking the link would open a second browser window.
- ▶ **Display**
This is mainly done using the Web page portlet which itself renders the external application into the portal page using an iFrame.
- ▶ **Integrated**
Mainly done by using existing portlets or low complexity development.
- ▶ **Migrated**
The external application is no longer used in its original state. All the functions of the user interface are reprogrammed in a portlet using the portal API and the API of the application to be integrated.

Table 9-1 Portlet pattern advantages/disadvantages

	Advantages	Disadvantages
Link	<ul style="list-style-type: none">▶ Quick and easy▶ Access existing applications▶ No modifications to existing functionality	<ul style="list-style-type: none">▶ Simple Web link from the portal▶ Doesn't use the portal framework to display the actual application
Display	<ul style="list-style-type: none">▶ Minimal enhancements to existing functionality▶ Optional link outside portal to access existing applications for more advanced functionality	<ul style="list-style-type: none">▶ Minimal or no application functionality within the portal framework
Integrated	<ul style="list-style-type: none">▶ Significant functionality within portal	<ul style="list-style-type: none">▶ Requires development time and might be more difficult to implement
Migrated	<ul style="list-style-type: none">▶ Full application functionality within portal	<ul style="list-style-type: none">▶ Technically challenging and resource intensive to implement

The corresponding levels of integration regarding LWWCM are shown in Table 9-2 on page 192.

Table 9-2 LWWCM/Content Manager integration pattern

Link	This is a link pointing to Content Manager content. This link can either be hard-coded or generated dynamically. You can do this with the federated component, connect tags or JSP component (see Chapter 7, “Publishing Content Manager content” on page 115). The content would normally open in a separate window once you click the link.
Display	<p>There are two ways to display external applications in portal:</p> <ol style="list-style-type: none"> 1. Usually this is done through the Web page portlet which offers the ability to pull an external Web page into an iFrame. Since this happens on the user’s machine, the user needs to have access to the remote URL. Any existing Web application displaying Content Manager content could be referenced in this iFrame. 2. Web clipping is another approach to display external Web applications in the portal. Technically, the WebSphere Portal Server server fetches the HTTP stream of a defined URL. Before it is actually displayed on the portal page, modifications can be done to the stream, such as URL rewriting, definitions of start and end points, login. Any existing Web application displaying Content Manager content can be referenced by the Web clipping portlet.
Integrated	This uses the existing LWWCM and Content Manager portlets provided by IBM. From an LWWCM point of view, it requires some development regarding portal-specific components and only a small effort to set up and configure the portlets. This is the preferred approach.
Migrated	This uses the API of both LWWCM and Content Manager to develop brand-new portlets. From an LWWCM point of view, there is no need to choose this method, since the level of integration you can reach with the existing portlets is sufficient in most cases. One example of this approach is the seamless integration of the LWWCM hierarchy into the portal navigation, instead of using the content viewer portlet as a navigator portlet. This can be achieved through a portlet which first uses the LWWCM API to read the hierarchical information and from then on uses the portal API to compose a native portal navigation structure.

In the following sections, we choose the integrated approach.

9.4 LWWCM portlets

There is only one portlet which is responsible for the integration of LWWCM into the portal. Only the configuration of this portlet makes the difference. There are three types of portlets which can be configured this way:

1. Dynamic content

A portlet of this type listens to events of other LWWCM portlet on the same portal page. For example: The content of this portlet changes dependent on the site area selected in another portlet.

2. Static content

A portlet of this type shows a fixed content which is independent from other LWWCM portlets on the portal page. This approach is generally used to show a logo or a banner.

3. Library component

Library components are components like tables, navigators, menus. Portlets configured to show this type of component are mostly used to present the user with some kind of navigator. Selecting an entry in the navigator triggers the event in dynamic content portlets (see above) to change the content.

9.4.1 Typical LWWCM portal page

The typical portal page integrating LWWCM would consist of two or three portlets. Portlet one shows the navigator, portlet two shows dynamic content depending on which entry is selected in the navigator. Optionally, portlet three could display a contextual menu. This is illustrated in Figure 9-2.

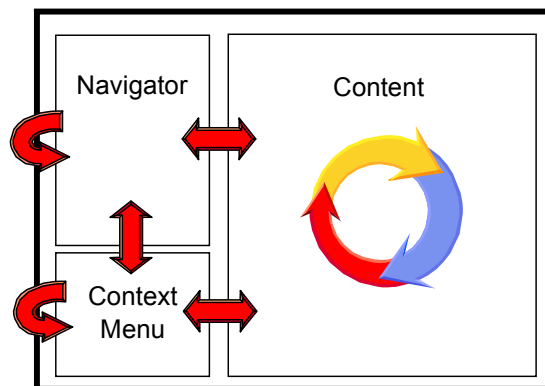


Figure 9-2 Sample LWWCM portal page layout

In the following sections, we create three portlets to show our insurance content within a portal page. The page will be made up of:

- ▶ A navigation portlet
- ▶ A dynamic content portlet showing content selected in the navigation portlet
- ▶ A static content portlet showing a menu

All three portlets will be merged into one portal page.

Navigation portlet

In this section, we discuss the creation of a content viewer portlet set up in such a way that it displays our left hand navigator. The top-level node is the insurance node.

First, we create a new navigator library component in LWWCM to have a navigator which starts displaying the hierarchy at the insurance node.

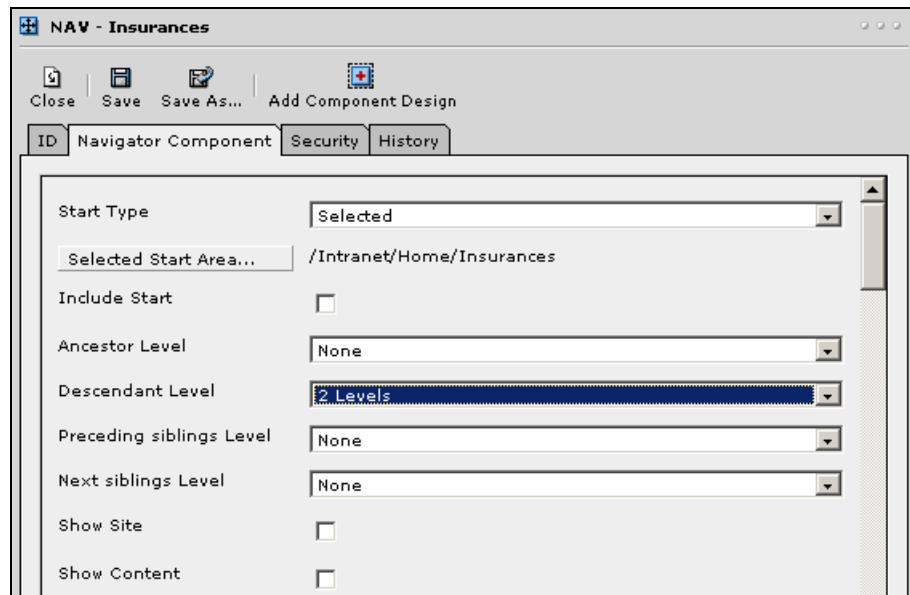


Figure 9-3 Insurance navigator definition

The navigator portlet is then configured to use this insurance navigator.

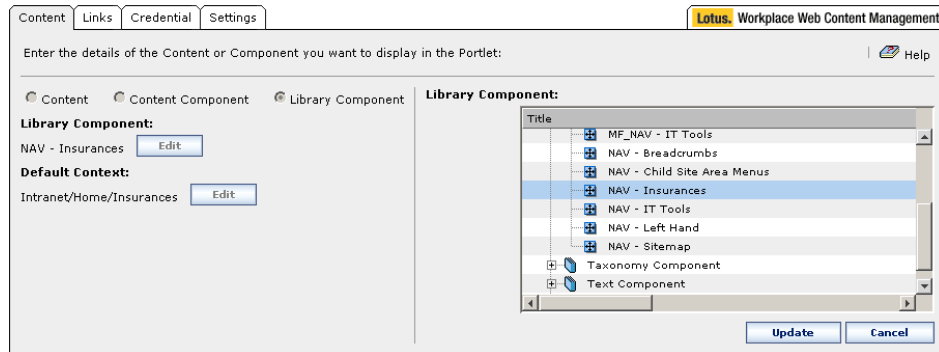


Figure 9-4 Insurance navigator portlet configuration - content

This portlet is broadcasting links to other portlet on the page to cause, for example, the content portlet to adopt context changes.

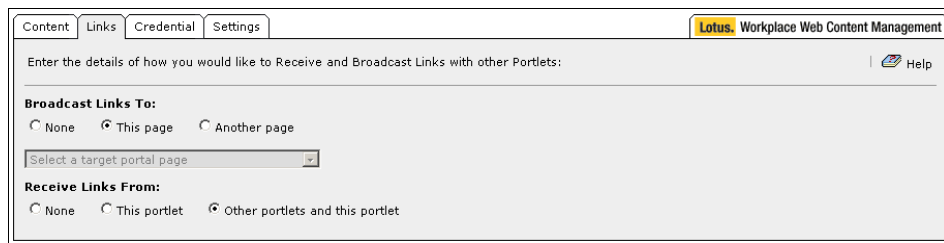


Figure 9-5 Insurance navigator portlet configuration - links

Content portlet

We provide the information on how to create the content portlet which is showing:

- ▶ LWWCM repository content
- ▶ External Content Manager content using the federated content component

For demonstration purposes, we create content for the site area called Insurance Name 1. This content also references the Content Manager content through an embedded federated component.

The content in Content Manager is created using the import function of the Content Manager client for Windows.

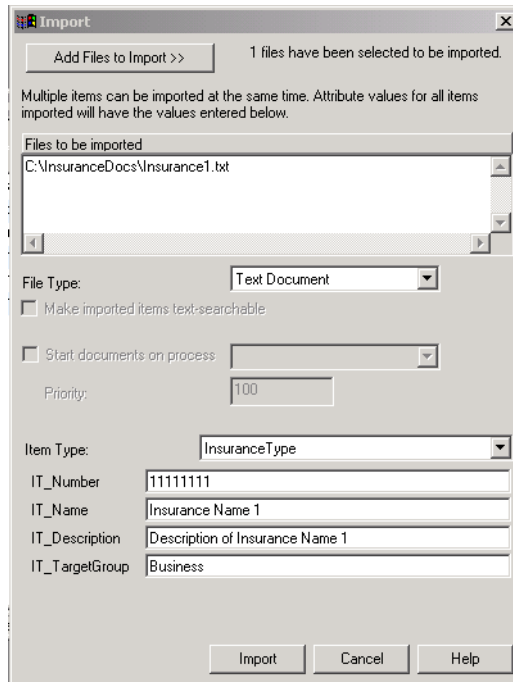


Figure 9-6 Importing text document into Content Manager

To keep it simple, we just create a copy of the existing content object IT, rename it to Insurance Name 1 and add a federated component to the content using the component manager named Insurance1Fed.

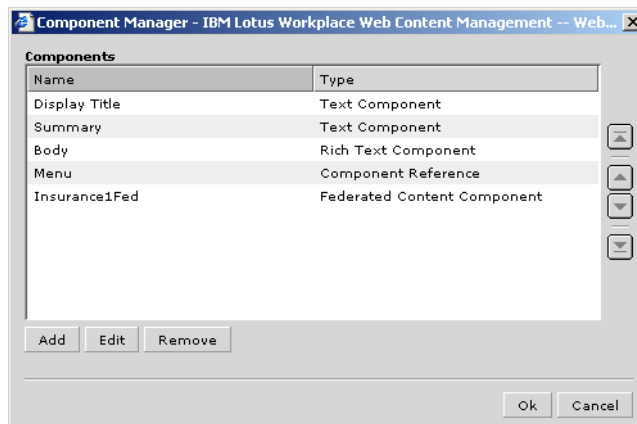


Figure 9-7 Component manager for content Insurance Name 1

After that, we open the new content in edit mode and configure the federated component within this document to reference the Content Manager content. Click **Select Reference** in the Insurance1Fed section in the content, enter your search criteria and select **Insurance Name 1** from the result set.



InName	InDesc	InNumber	InGroup
Life Insurance	Life Insurance Detailed Description	00000001	Personal
Automobile Insurance	Automobile Insurance Detailed Description	00000002	Personal
Longterm Care	Longterm Care Insurance Detailed Description	00000003	Personal
Investment Mgmt	Investment Management Detailed Description	10000001	Business
Retirement Program	Retirement Program Insurance Detailed Description	10000002	Business
Insurance Name 1	Description of Insurance Name 1	11111111	Business

Figure 9-8 Selecting content for the federated content component

We then also change the necessary fields such as DisplayTitle, Summary and Body. In order to see the new fields and the content on the page, the LWWCM page design which is going to be used on the portal page (here: PortalPageLayout) must contain additional lines as shown in Example 9-1.

Example 9-1 Additional lines in PortalPageLayout

```
<b>Insurance Name:</b> <FederatedCmpnt AttributeName="InName" context="current"
key="Insurance1Fed" Scope="attribute" type="content"/><br>

<b>Insurance Description:</b> <FederatedCmpnt AttributeName="InDesc"
context="current" key="Insurance1Fed" Scope="attribute" type="content"/><br>

<b>Insurance Number:</b> <FederatedCmpnt AttributeName="InNumber"
context="current" key="Insurance1Fed" Scope="attribute" type="content"/><br>

<b>Insurance Target Group:</b> <FederatedCmpnt AttributeName="InGroup"
context="current" key="Insurance1Fed" Scope="attribute" type="content"/><p>

<b>Content:</b><br><FederatedCmpnt context="current" key="Insurance1Fed"
Scope="content" type="content"/>
```

Note: See also Chapter 7, “Publishing Content Manager content” on page 115 for more details about how to reference a federated component in a page design.

After the creation of the content in LWWCM, we configure the content portlet to use this new content.

Content

Links

Credential

Settings

Lotus. Workp

Enter the details of the Content or Component you want to display in the Portlet:

☒ Content
 ☐ Content Component
 ☐ Library Component

Content Component:
 Intranet/Home/Insurances/Insurances

Alternative Page Design:
 PortalPageLayout

Content:

Title

+ Sales & Marketing

+ IT

+ Thank-you

[-] Insurances

+ Insurance Type 1

+ Insurance Type 2

+ Insurance Type 3

[-] Insurances

+ Tools

The chosen content will be overwritten with new content when a link in the navigator is clicked, since this portlet is configured to listen to broadcasts from other portlets and itself.

Content	Links	Credential	Settings	Lotus. Workp
---------	-------	------------	----------	--------------

Enter the details of how you would like to Receive and Broadcast Links with other Portlets:

Broadcast Links To:

☐ None
 ☒ This page
 ☐ Another page

Receive Links From:

☐ None
 ☐ This portlet
 ☒ Other portlets and this portlet

Menu portlet

The menu portlet is created to show a menu. This menu shows a list of available insurances and is retrieved from Content Manager using the JSP component created in 7.5, “JSP component” on page 151. The name of the JSP component used here is MyJSPCmpnt.

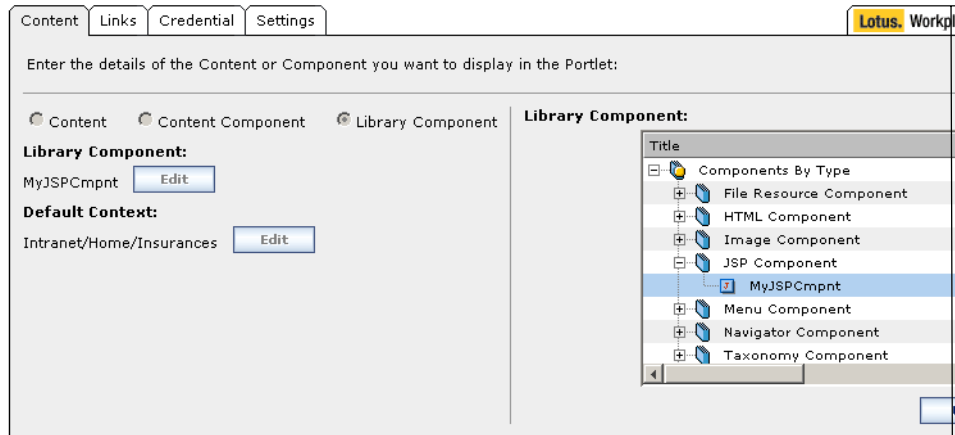


Figure 9-11 Insurance menu portlet configuration - content

Creating the portal page

Once all the portlets are set up, they can be put on a portal page.

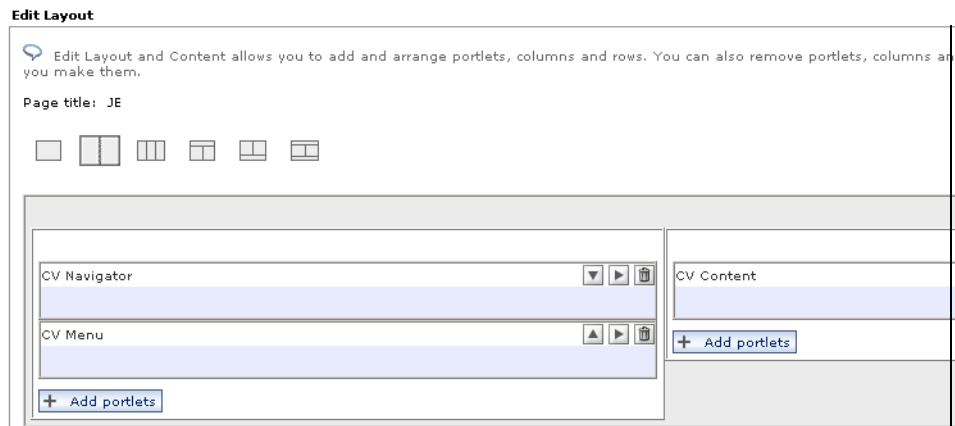


Figure 9-12 Portal page layout

The resulting page should look similar to the following screen shot.

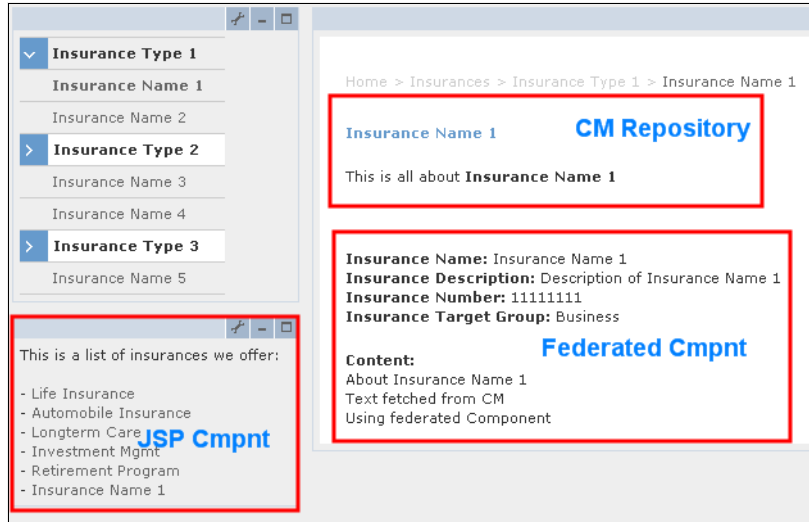


Figure 9-13 Insurance portal page

9.4.2 User management and security

The major challenge when trying to integrate Content Manager content through LWWCM into a portal page is that there are three different systems and all of them claim to have their own user management. In the portal context, the user has to log in to the portal first. In order to prevent users from having to enter their credentials a second time, the user information has to be passed through to LWWCM. This can be done by configuring each portlet on the credentials tab in such a way that it either inherits the portal user credentials or uses a credential vault slot.

Note: For more information regarding the setup of credential vault slots, see the WebSphere Portal Information Center.

In both cases, the portal authenticates with a valid user/password combination against LWWCM. This ensures that the logged-in user will only be able to see LWWCM components and content which he/she is allowed to see. LWWCM itself does not pass the credentials through to Content Manager. Instead, it connects to Content Manager with one global user defined in the Content Manager persistency section of [LWWCM_HOME]\config\aptrixjpe.properties. In general, this user is granted at least read access to all objects in Content Manager:

```
cm.username=icmadmin
cm.password=password
```

The security is implemented in LWWCM and not in Content Manager.

The flow described above is illustrated in Figure 9-14.

With LWWCM V2.0, the user management is not yet integrated into the WebSphere Member Manager (WMM) which is the basis for WebSphere portal server. LWWCM is still running its own user management system. But there are ways to realize a pseudo-integration. This is discussed in Chapter 11, “Security” on page 237.

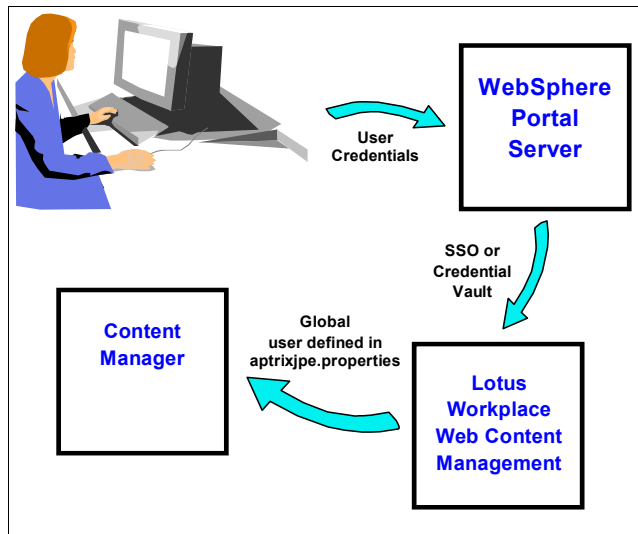


Figure 9-14 User credentials flow

9.5 Content Manager portlets

We will not discuss the Content Manager portlets in detail here since they are not directly related to the integration of LWWCM and Content Manager. An approach could be to put LWWCM portlets on a portal page and to complement them with Content Manager portlets wherever LWWCM cannot provide the needed functionality. The Content Manager portlets and LWWCM portlets remain independent though, that is, for example, they do not communicate using click to action. Therefore, the preferred approach is to have all your needs covered by either the LWWCM *or* the Content Manager portlets.

In the following sections, we give a rough overview of available portlets for Content Manager and Information Integrator for Content.

9.5.1 Content Manager/Information Integrator for Content portlets

These portlets perform the following on Content Manager content:

- ▶ Read functions similar to the eClient Web application and the Content Manager administration client for Windows.
- ▶ Import new content to Content Manager.
- ▶ Export existing content from Content Manager.
- ▶ Edit attributes.

You can download Content Manager portlets from the IBM portal and workplace catalog.

1. Go to <http://catalog.lotus.com/wps/portal/portalworkplace>.
2. Enter IBM Content Manager into the search field and press **Enter**.
3. Look for Portlets for IBM DB2 Content Manager V8.2.
4. Download the war file and the documentation.

The following screen shots give a basic idea of what the Content Manager portlets look like.

Figure 9-15 shows the start page of the portlet which gives you the option to either run a query on Content Manager content, create new folders or import documents into Content Manager.

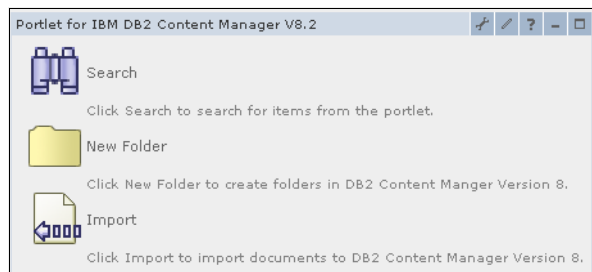


Figure 9-15 Content Manager portlet start page

Figure 9-16 on page 203 shows the page where all available item types are listed.

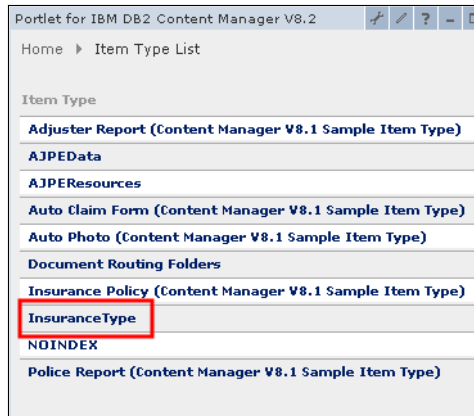


Figure 9-16 Content Manager portlet items types

Figure 9-17 shows the page where you can enter your query.

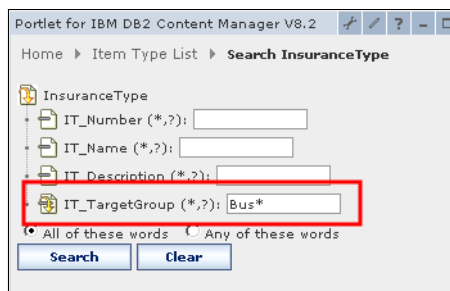


Figure 9-17 Content Manager portlet search entry

Figure 9-18 on page 204 shows the result page of a search.

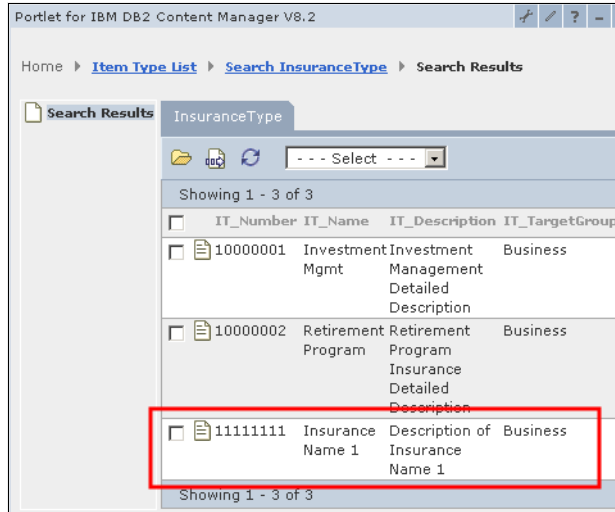


Figure 9-18 Content Manager portlet search result

9.5.2 Information Integrator for Content portlets

These portlets perform the following on Content Manager content:

- ▶ Federated Search performs a similar function to the EIP thin client application. It presents you with a list of predefined search templates, allows you to enter search parameters, and displays search results in a table.
- ▶ Advanced Search enables you to perform an unstructured search using Information Integrator for Content's Information Mining feature. You can select from a list of categories to narrow the search results.

You can download Information Integrator for Content (former name was EIP) portlets from the IBM portal and workplace catalog.

1. Go to <http://catalog.lotus.com/wps/portal/portalworkplace>.
2. Enter IBM EIP into the search field and press **Enter**.
3. Look for IBM EIP Search Version 8.1.
4. Download the war file and the documentation.

The screen shots below (Figure 9-19 on page 205 and Figure 9-20 on page 205) show an example of a federated search and the result set.

WebSphere Portal Welcome wpsadmin!

Home

Welcome

EIP Federated Search

Template:
WpsOnAlamo

LName:
LIKE %

fName:
LIKE

SSN:
LIKE

Search

Figure 9-19 Federated search portlet - query screen

EIP Federated Search

Search Results: [New Search](#) [Refresh](#)

	Source	Userid	Time Stamp
View	IMPORT	USER2	2002-11-05-18.30.09.894000
View	IMPORT	USER2	2002-05-01-16.00.16.222000
View	IMPORT	USER2	2002-05-16-17.06.39.455000
View	IMPORT	USER2	2002-11-15-11.41.17.939000
View	IMPORT	USER2	2002-08-27-14.41.53.959000
View	IMPORT	USER2	2002-08-27-14.42.44.421000
View	IMPORT	USER2	2002-03-12-12.27.23.395000
View	IMPORT	USER2	2002-04-11-17.44.15.565000
View	IMPORT	USER2	2002-11-05-18.21.23.597000
View	IMPORT	USER2	2002-12-03-10.44.43.234000
View	IMPORT	USER2	2003-02-05-13.51.50.145000
View	IMPORT	USER2	2002-03-07-13.10.00.567000
View	importedFromPClient	zeroLenghtGif	01234567890
View	IMPORT	USER2	2002-12-03-14.02.56.093000
View	IMPORT	USER2	2002-09-09-15.29.35.976000
View	IMPORT	USER2	2002-03-12-12.26.39.512000
View	IMPORT	USER2	2002-03-07-13.11.20.782000
View	IMPORT	USER2	2002-03-22-14.32.13.152000

Figure 9-20 Federated search portlet - result screen



Part 3

Advanced topics

In this part, we cover advanced topics such as search, security, integrating VideoCharger videos, data migration, and versioning.



Search

This chapter describes how to search Content Manager text content and how to integrate the search results in a Web page or portal with LWWCM. We describe different search approaches, and the advantages and disadvantages of using:

- ▶ Net Search Extender (NSE)
- ▶ Federated search with Information Integrator for Content (II for Content)
- ▶ LWWCM integrated search
- ▶ Portal Search Engine (PSE)
- ▶ Domino Extended Search (DES)

The Information Integrator for Content and NSE approaches will be discussed in more detail.

10.1 Overview

Information searches on the Web have become an elementary part of our daily life. The question is how to find the right information among all that is available on the Web. The question also translates into what search engine we should use to find the right information, assuming that a good search engine helps us to find what we are looking for. A good search engine might not be the universal answer. This answer depends on what we are looking for: do we just want to browse the Web to see what information is out there or do we need specific information? It also depends on where we are and how much time we have: are we searching at work, at home, or on a business trip? Do we have to find the information within a short time frame or not? Some of the differences between the search engines on the market are the amount of information they index, the ranking approaches of the search results, how the results are displayed, whether there is duplicated information, metadata search to different search engines or back-ends. Intelligent search solutions include summarization, categorization, fuzzy searches, using thesaurus dictionaries, redefining the query based on the retrieved results list, influencing the ranking order, and including Web crawlers which can be configured by users. Other important features of a search input panel are the possibility for simple and advanced searches, personalized searches, word or phrase searches, boolean searches with AND, OR, NOT, proximity searches in the same sentence or in the same paragraph, similarly spelled search terms, degrees of fuzziness, free text searches and wildcard searches on characters or word levels.

From the Enterprise content management point of view, searches are a part of every solution. Content is nothing if one cannot find it and retrieve it quickly, regardless of the back-end server in which the content is stored.

10.1.1 Content Manager search possibilities

Content Manager has different ways of finding and retrieving information stored within it: parametric searches, full text searches, and combinations of parametric and full text searches.

A *parametric search* is based on the Library Server metadata. In our case study example, the metadata includes an attribute field, IT_Name, in the insurance item type definition, representing the insurance type name. After importing the item objects with the Content Manager Windows client or your own application, you can run a basic or advanced parametric search on the attribute field, IT_Name, whereby you specify the full attribute entity you search for. Figure 10-1 on page 211 shows a sample input window for a basic search and Figure 10-2 on page 211 shows the sample search results.

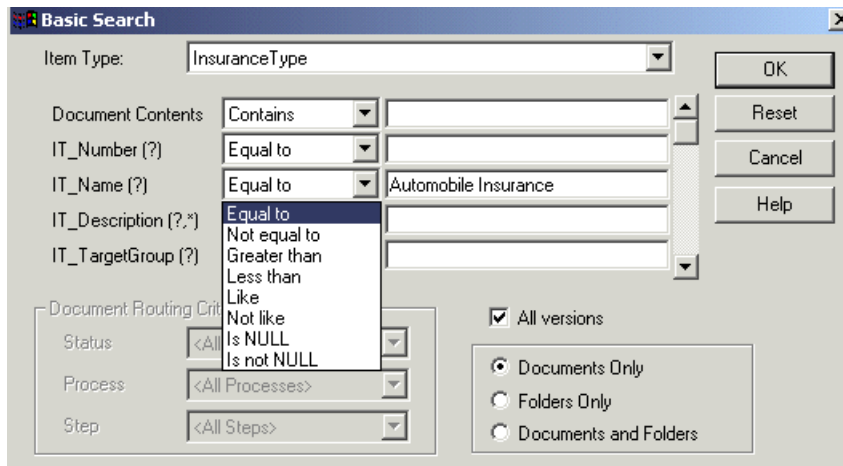


Figure 10-1 Parametric search on the InsuranceType attribute IT_Name

Version	IT_Number	IT_Name	IT_Description	IT_TargetGroup	IT_Duration	IT_Price
1	00000002	Automobile Insurance	Automobil...	Personal		
1	00000004	Automobile Insurance	Automobil...	Personal	1 year	1000.000000
2	00000002	Automobile Insurance	Automobil...	Business		
1	00000005	Automobile Insurance	Automobil...	Personal	2 years	1700.000000
1	00000008	Automobile Insurance	Automobil...	Personal	5 years	3000.000000

Figure 10-2 Parametric search results

A *full text search* is based on document and attribute indexes. For item type attributes or item type text document objects that have been set up to be text indexed, this search type can be used. A full text search is performed against the NSE text index instead of the Library Server attribute tables.

A combination of parametric search and full text search is the third search possibility available in Content Manager; it combines the means of parametric and full text searches in one query string. Which search option can be used is defined during the item type definition.

The following is a list of some important factors related to Content Manager search possibilities:

- ▶ When you want to make an item type (or text document) text searchable, you need to index the text part of your document. Make sure you add *ICMBASETEXT* as a part type of the item type in the Content Manager System Administration Client's Document Management tab.

ICMBASETEXT contains the fundamental content of a document item type that stores text content. If you plan to index a text part of your document, you should store the part in this part type. Indexing a text part allows a text search to be performed on the content of the part.
- ▶ *ICMBASE*, the base document part, stores any non-textual type of content such as image and audio. To ensure that an item is viewable from eClient, the item type of the item must include at least one base document part.
- ▶ From the Content Manager Windows client, you can run *advanced searches*, a powerful feature. Using an advanced search, you can specify groups for several item types and set different criteria for every one of them. For each item type, you create a search expression using the search criteria (the attribute values) for that item type.

10.1.2 NSE integration within Content Manager

With NSE installed, you can make Content Manager attributes, resource items and documents text searchable. The text search capability can be enabled when creating a new item type.

For an overview of Net Search Extender, see 2.2.10, "IBM DB2 UDB Net Search Extender (NSE)" on page 21. For more in-depth information, refer to *Net Search Extender Administration and User's Guide*, SH12-6740.

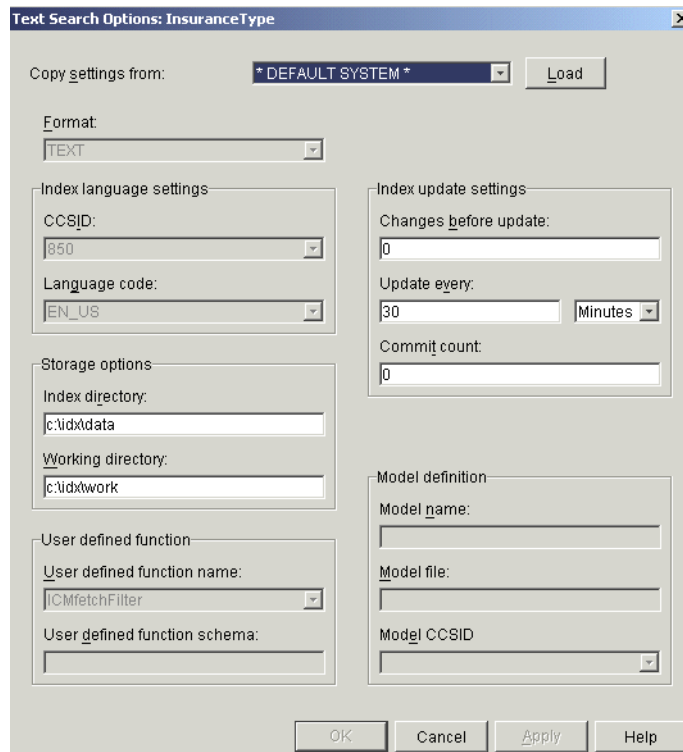
Note that, by default, the Content Manager item types *AJPEData* and *AJPEResources* are not text searchable. These item types are used to store LWWCM system data information such as file resource components and image components. There is no reason for the Web content development perspective to enable a text search on these item types.

Integrating search results in Web content

An interesting and useful approach is to dynamically integrate the Content Manager text search results into Web content. It is also interesting and useful to integrate a JSP that contains the Content Manager text search function in a Web page so that the user can dynamically run the text searches against the Content Manager's text-searchable content.

Defining Content Manager text search options

Text search options can be set from the Content Manager System Administration Client when creating an item type. During the creation of a new item type, at the New Item Type Definition panel for Document and Resources item type classification, click **Enable text search** and then click **Options**. This allows you to go to the text search option setting window (see Figure 10-3) to set the various options.



The dialog box titled "Text Search Options: InsuranceType" contains several sections for configuring text search parameters. At the top, there is a "Copy settings from:" dropdown menu set to "* DEFAULT SYSTEM *" and a "Load" button. Below this is a "Format:" dropdown menu set to "TEXT". The "Index language settings" section includes a "CCSID:" dropdown menu set to "850" and a "Language code:" dropdown menu set to "EN_US". The "Storage options" section includes an "Index directory:" text field with "c:\idx\data" and a "Working directory:" text field with "c:\idx\work". The "User defined function" section includes a "User defined function name:" dropdown menu set to "ICMfetchFilter" and a "User defined function schema:" text field. The "Index update settings" section includes a "Changes before update:" text field set to "0", an "Update every:" section with a text field set to "30" and a dropdown menu set to "Minutes", and a "Commit count:" text field set to "0". The "Model definition" section includes a "Model name:" text field, a "Model file:" text field, and a "Model CCSID" dropdown menu. At the bottom, there are "OK", "Cancel", "Apply", and "Help" buttons.

Figure 10-3 Text search option settings

We only need to consider setting the main parameters. Figure 10-3 shows the values we set for item types in our case study. We use the default for most of the values. You can set them according to your environment:

- ▶ In the index language setting fields, specify a supported *code page* (CCSID) and *language code* for the documents or attribute fields that will be used to create the text index. By default, we use 850 for CCSID and EN_US for Language code.
- ▶ In the Model definition fields, you can specify parameters for a model that describes what sections of the text are to be indexed. In our case study sample, we leave the fields blank.

- ▶ In the Index update settings fields, you can specify the amount of time that needs to have passed between updates of the text indexes. Updating text indexes ensures that the text search results reflect the latest changes on the documents or attributes enabled for text search. How frequently you want to set these values depends on how often the data is changed and on your system resources. For example, if your documents are only updated once per week, you do not need to set the updates to every hour. To give another example, if versioning is enabled so that a new version will always be created, then the update frequency you choose depends on how often the new versions are created. Be aware that the amount of time that passes before a text index update impacts your system performance. You may also choose to specify the number of changes to an item type object before the next update runs in this setting. In our case study, we set values so as to update the index every 30 minutes. This is because we have a small testing volume and want the indexing to take place frequently.
- ▶ In the User defined function fields, you can choose between two user-defined functions, *ICMfetchContent* and *ICMfetchFilter*:
 - *ICMfetchContent* supports the following document formats: TEXT for plain text (for example, flat ASCII files), HTML for Hypertext Markup Language, XML for Extended Markup Language, and GPP for General Purpose Format (flat text with user-defined tags).
 - *ICMfetchFilter* supports filtering software to extract textual content from PDFs and other common text formatting tools, such as Microsoft Word. The libraries are not part of Net Search Extender and need to be installed separately as described in the NSE documentation.

In our case study, we use the default value *ICMfetchContent*.

Note: If you update an item type or attribute from non text-searchable to text-searchable, only the new imported text documents or newly specified attributes will be indexed for text search, not the existing ones. Once the Content Manager text search options for an item type have been defined, they cannot be changed.

Importing text-searchable item objects into Content Manager

To import text-searchable item objects into Content Manager, you can use the Content Manager Windows client, eClient, or your application. Within the Content Manager Windows client, you must specify that the document is text searchable when importing the document to Content Manager. With the eClient, the document is always imported as text searchable if the text search option was enabled during item type definition. Make sure you specify the right content and file type before you import a text document. See Figure 10-4 on page 215 for an example of importing a document using eClient.

Import Document - Microsoft Internet Explorer

Import Document ?

Server : DEMO2DB

Item type : InsuranceTyp

* denotes a required field

IT_Name : Policy2341

IT_Description : House insurance policy

* File type : TEXT

* Content type : Textual document

* File : C:\policy2341.txt Browse...

Import Close

Figure 10-4 Import a text searchable document form Content Manager eClient

10.1.3 Federated search

Federated search enables users to search information from different sources. Enterprise data sources are diverse and distributed. About 80% of the data is in unstructured format. Information Integrator for Content's federated search engine centralizes the searches from within the Content Manager systems and the other distributed content sources and repositories. A federated search also includes the retrieval of free text capabilities by using NSE as in our installation or as part of Information Mining. Information Mining is a feature that enables users to search among unstructured data. It includes categorization, clustering, summarization, and feature extraction for the unstructured data.

Integrating a federated search in Web content

In 10.1.4, "LWWCM search module" on page 219, we show how the federated search can be integrated during the Web site development phase using Content Manager as a repository. You create the federated search templates within the Information Integrator for the Content administration client. In this example, the *federated search template* goes against the Content Manager V8 repository only. It is possible to create a federated search template against multiple and different back-end content repositories. From the LWWCM administration client, you can create *federated content components* referring to the created federated search template. Once the federated content components are created, the LWWCM Web developers can integrate the federated search results into an LWWCM Web

page content. See 7.3, “Federated content component” on page 123 for the steps to create the federated content component.

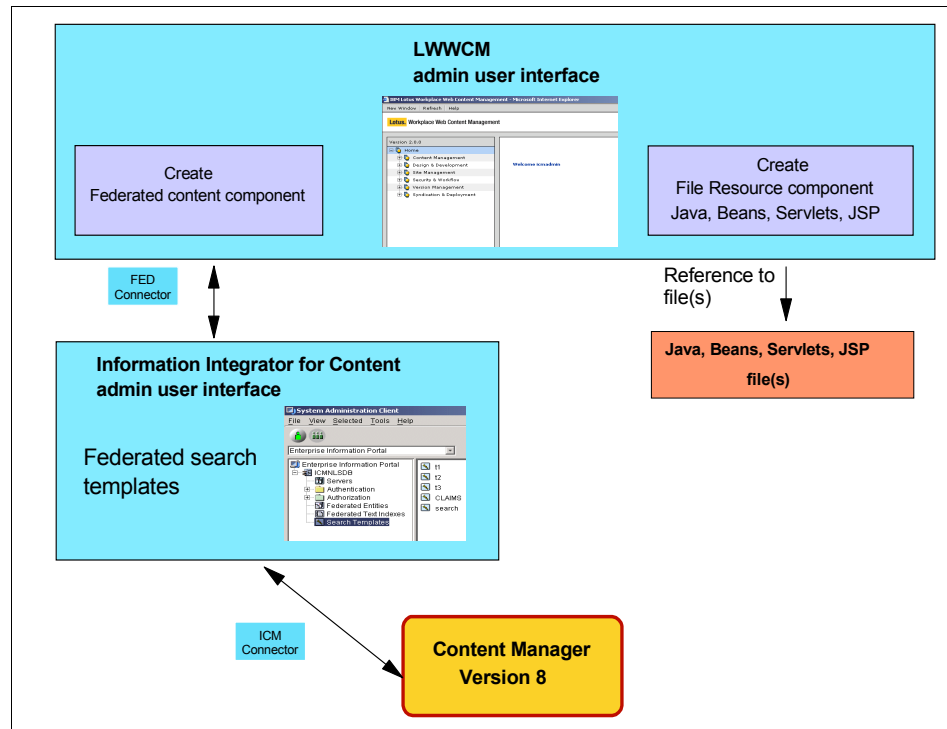


Figure 10-5 LWWCM Web site development with federated search objects

An even more flexible approach is to integrate JSP resource components that use Java search APIs as described in 10.2, “Integrating a search” on page 219. With JSP incorporating federated searches in a Web site, users have the possibility to query dynamically against the back-end data repository supported by the federated search in Information Integrator for Content. In our case study scenario, as shown in Figure 10-6 on page 217, users can use a federated search to query against Insurance Name, Insurance Description, Insurance Group, and Insurance Number. These federated attributes are mapped to Content Manager attributes, so the federated search is against the Content Manager data repository. The Web page shown in Figure 10-6 on page 217 also displays Content Manager metadata and content objects.

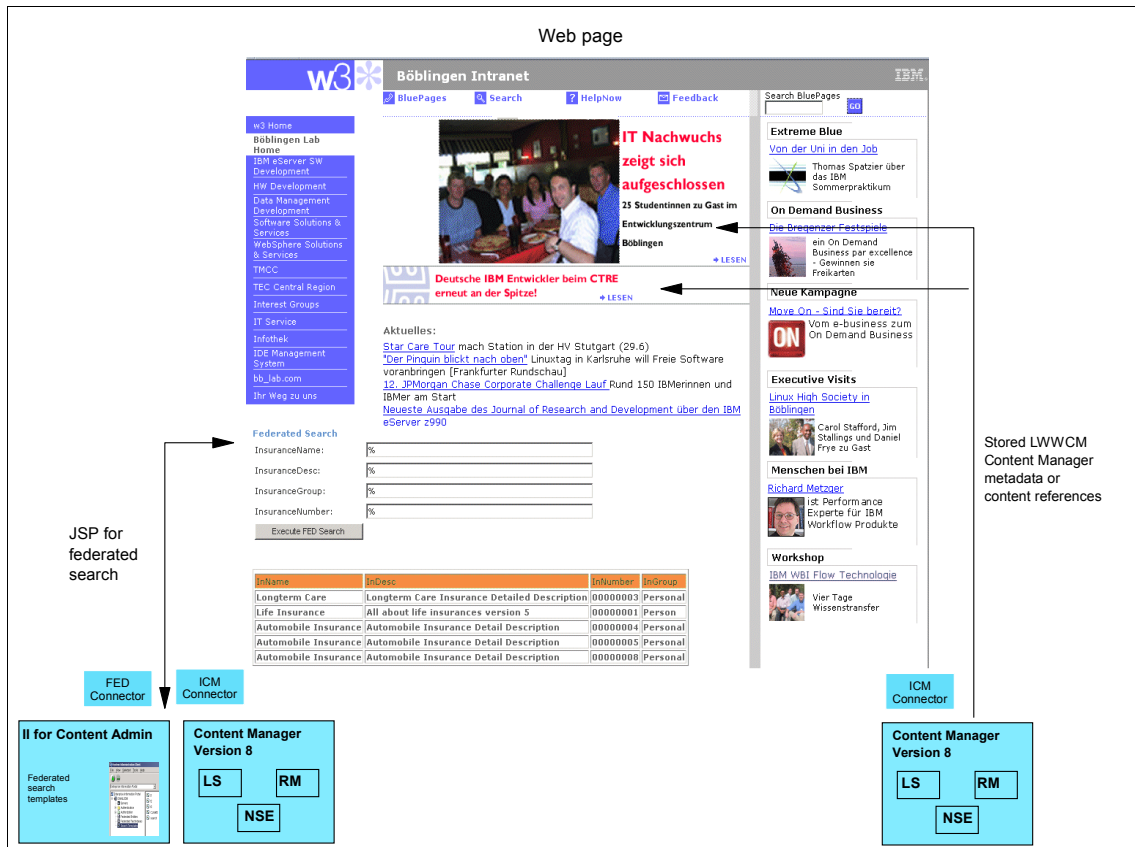


Figure 10-6 JSP with federated search APIs integrated in a Web site

To become familiar with the federated search APIs syntax, see the JSP search sample in the Information Center subdirectory:

- ▶ On AIX, it is /usr/lpp/cmb/samples/jsp/taglib
- ▶ On Windows, it is <CMCOMMON>\infoctr\info\apiinfo\dk\samples\jsp\taglib

From the Information Integrator for Content (also known as EIP) Development window, you can compile and run the sample files such as TSearch.java. In Windows, you can find this file in the <CMCOMMON>\infoctr\info\apiinfo\dk\samples\java\beans directory. With TSearch, you can run federated or Content Manager item searches from the command line. See the syntax and a sample TSearch request, including the query results, in Example 10-1 on page 218. This example queries the InsuranceVideo item objects.

Before you run the sample, make sure you have the right environment by starting with the EIP Development Window or running cmbicmenv81.bat to run a Content Manager search or cmbenv81.bat (cmbenv81.sh on AIX) to run a federated search. These batch files automatically set the classpath for all Java utilities and samples. Also, edit the files and, if necessary, change the settings such as DB2HOME or CMBROOT according to your environment setting.

Example 10-1 Sample TSearch query for InsuranceVideo

```
C:\ibm\cmbroot\cmbenv81. bat
C:\IBM\Cmgt\infoctr\info\apiinfo\dk\samples\java\beans>javac TSearch.java
C:\IBM\Cmgt\infoctr\info\apiinfo\dk\samples\java\beans>java TSearch
Searches a server for items, using a query string.
```

Syntax:

```
java TSearch <dstype> <server> <userid> <password> <entity> <condition>
where:
  <dstype> is the type of server (Fed, ICM, DL, OD, DB2)
  <server> is the name of the server or database. This can be
            followed by a connect string in parenthesis.
  <userid> is the userid
  <password> is the password
  <entity> the name of the entity (aka item type, table) to search.
  <condition> the condition portion of the query string. The syntax
              of the condition varies depending on server type.
```

Examples:

```
java TSearch ICM icmnlldb icmadmin password Book @Title LIKE \"%\"
java TSearch DL libsrvrn frnadmin password NOINDEX Userid LIKE '%'
java TSearch Fed cmbdb cmbadmin password NoIndex Userid LIKE '%'
java TSearch DB2 sample db2admin password EMP_PHOTO PHOTO_FORMAT LIKE 'gif'
java TSearch OD odserver userid password Balance CustName like '%BLONDIE%'
```

```
java TSearch ICM icmnlldb icmadmin password InsuranceVideo @IT_Name LIKE \"%\"
```

```
Connecting to server
```

```
/InsuranceVideo[ @IT_Name LIKE "%"]
```

```
Performing search
```

```
There are 2 items found.
```

```
Item[0] = Life insurance - Families
```

```
IT_Name: Life insurance - Families
```

```
IT_Description: Video for policy Life23 Second Version
```

```
IT_Number: Life23
```

```
IT_Group: Person
```

```
SOURCE: Video from marketing team
```

```
Item is a document.
```

```
Item is not checked out.
```

```
There are 1 content parts.
```

```
Content Part[0] = length: 8563944 mimetype: video/mpeg
```

```
There are 0 annotation parts.
```

```
There are 0 notelog parts.
```

```
Item[1] = Life insurance - Car driver
```

```
IT_Name: Life insurance - Car driver
IT_Description: Video for policy Life43 First Version
IT_Number: Life3
IT_Group: Person
SOURCE: Video from department 3421
Item is a document.
Item is not checked out.
There are 1 content parts.
  Content Part[0] = length: 809599 mimetype: video/mpeg
There are 0 annotation parts.
There are 0 notelog parts.
Disconnecting from server
```

To integrate (embed) JSP within LWWCM, see 7.5, “JSP component” on page 151. For a simple integrated JSP search sample with LWWCM, see 7.5.4, “Sample code” on page 156.

10.1.4 LWWCM search module

The LWWCM search module can be used to create a basic search function on keywords in objects stored in a Web Content Management environment on a limited scale. Queries may be composed of keywords combined with AND/OR operators to create a more specific query. For details, see 10.2.3, “LWWCM internal search” on page 232.

10.2 Integrating a search

In this section, we discuss the different approaches to providing users with an adequate search front end to query both content objects stored in the LWWCM repository and external content resources. There is not *the* search approach which would cover all requirements; but with some extra effort, you could present users with only one search interface which forwards the query to the appropriate search engine. The search approaches discussed here include:

- ▶ Leveraging the Content Manager/Information Integrator for Content Java APIs, which include:
 - Using *Net Search Extender (NSE)* from within a LWWCM Web page to query Content Manager content text attributes and text resources.
 - Using *Federated Search* from within a LWWCM Web page to run queries against different sources such as DB2, Domino Doc and others. In our sample, we query Content Manager attributes.
- ▶ Using the *internal LWWCM search engine* to query content objects stored in the LWWCM repository.

- Pulling an existing search interface into the LWWCM Web page through the use of *iFrames*.

10.2.1 Leveraging the CM/II for Content Java APIs

To demonstrate how to leverage the Content Manager and Information Integrator for Content Java APIs, we use our case study to describe how to set up a sample search page for insurance types on your LWWCM Web page. We show you how to create a user interface to enter the query as well as a result table where clicking a result will open the actual document stored in the Resource Manager of Content Manager.

If you are not familiar with our case study, refer to Chapter 4, “Case study” on page 41. If you have not created the necessary Content Manager data model for the case study, refer to 7.2.3, “Content Manager data” on page 118 to see what item types need to be created and how they can be created.

Restriction: When using a federated search to search for Content Manager content, remember that it can only cover the metadata (attributes) of item types.

The steps to integrate NSE or the federated search are similar:

1. Develop the JSP file responsible for taking the query attributes, executing the search and displaying the results.
2. Create the necessary LWWCM objects using the LWWCM administration application:
 - a. Create a new Content Template to incorporate the JSP component.
 - b. Create a new HTML component to present the search entry form.
 - c. Create a new Page Design to place the search entry form and the JSP component on the page.
 - d. Create new content using the template created in Step a.
 - e. Create a new site area below the Tools site area and assign the newly created content as the default content.

Content Manager (integrated NSE) and Information Integrator for Content (for a federated search) provide a set of powerful and comprehensive Java APIs which can be programmed to perform almost any desired functionality in Content Manager. The presented approach leverages the new JSP component of the LWWCM framework to access these APIs from within LWWCM. The code is using the ICM/FED connector to connect to Content Manager and/or Information Integrator for Content server, execute the query and retrieve the results. The *advantage* of using NSE is that it cannot only search in attributes but also within the attached Content Manager content if this is enabled for the searched item

type. The *disadvantage* of using NSE compared to the federated search is that NSE can only search within Content Manager content whereas the federated search can search among many other different sources such as DB2, and Domino Extended Search.

Tip: Using the JSP component is also discussed in greater detail in 7.5, “JSP component” on page 151.

Figure 10-7 shows the theoretical composition of an LWWCM Web page which incorporates a text search using NSE. Using the federated search for integration works in the same manner and is discussed in 10.1.3, “Federated search” on page 215.

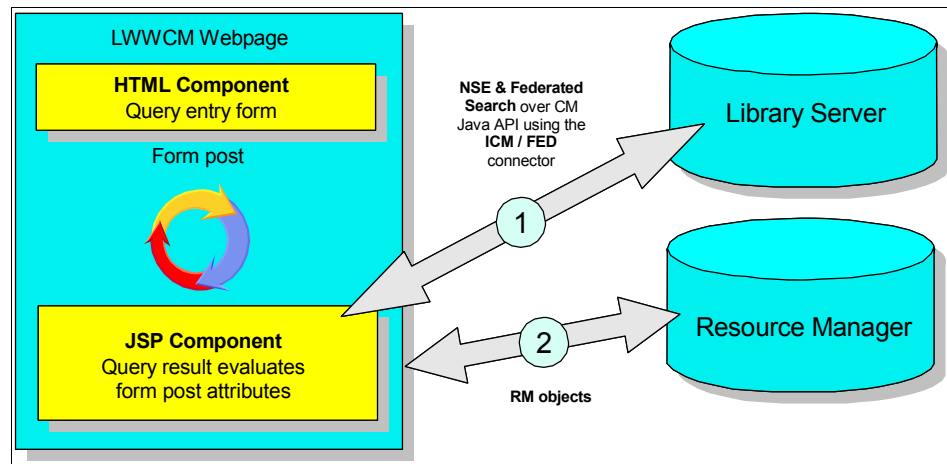


Figure 10-7 NSE / federated search integration

Developing the JSP file

We present sample codes for the NSE JSP file and federated search JSP file.

The Net Search Extender (NSE) JSP file

The sample code in Example 10-2 on page 222 reads the query string from the request and executes a text search query against the Content Manager server. After that, it loops through the result set and displays any available attributes of the found Content Manager content documents.

Copy and paste the sample JSP file in Example 10-2 on page 222 to
<WEB_APP_HOME>\jsp\NSESearch.jsp.

Note:

- The *servertype* is *ICM*.
- The *query* syntax is XPATH and contains text search parameters. For demonstration purposes, the query is currently constructed to search within attached Content Manager content. The query could easily be enhanced to cover an attribute search as well. The following example would also cover an attribute search:


```
String query = "/InsuranceType[contains-text-basic(./ICMPARTS/@TIEREF, \"\" + q[0] + \"\")=1 AND @VERSIONID = latest-version(.) OR @AttributeName LIKE \"%xyz%\""]";
```
- The *query* is constructed in such a way that it only retrieves the latest version of any document matching the query. This is done by adding `@VERSIONID = latest-version(.)` to the query string.

Example 10-2 NSESearch.jsp

```
<%@ page import="com.ibm.mm.beans.*, java.net.URL"%>
<%
// -----
// Read the posted query parameter
// -----
String q[] = request.getParameterValues("query");

// -----
// Define connection parameters of type ICM
// -----
String servername = "ICMNLSDDB";
String servertype = CMBBaseConstant.CMB_DSTYPE_ICM;
String userid = "icmadmin";
String password = "password";
String query = "/InsuranceType[contains-text-basic(./ICMPARTS/@TIEREF, \"\" + q[0] + \"\")=1 AND @VERSIONID = latest-version(.)]";

// -----
// Connect
// -----
CMBConnection conn = new CMBConnection();
conn.setServerName(servername);
conn.setDsType(servertype);
conn.setUserid(userid);
conn.setPassword(password);
conn.connect();

// -----
// Run the XPATH query
```

```

// -----
CMBQueryService qs = conn.getQueryService();
qs.setAsynchSearch(false);
qs.setQueryString(query, CMBBaseConstant.CMB_QS_TYPE_XPATH);
qs.runQuery();

// -----
// Get the resultset and loop over it
// -----
CMBDataManagement dm = null;
CMBSearchResults result = new CMBSearchResults();
result.setConnection(conn);
result.newResults(qs.getResults());

%><TABLE border=1><%

for (int i = 0; i < result.getCount(); i++)
{
%><TR><%
    CMBItem item = result.getItem(i);

    // -----
    // Header Row
    // -----
    if (i == 0) {
%> <TR bgcolor=FFF000><%
        for (int y = 0; y < item.getAttrNames().length; y++)
        {
%>         <TD><%= item.getAttrName(y) %></TD><%
        }
%> </TR><%
    }
    dm = conn.getDataManagement();
    dm.setDataObject(item);
    URL url = dm.getContent(0).getDataURL();
    for (int y = 0; y < item.getAttrValues().length; y++)
    {
%>         <TD><a href="<%= url.toString() %>" target="Result"><%=
item.getAttrValue(y) %></a></TD><%
    }
%></TR><%
}
%></TABLE><%

```

```
// -----
// Disconnect
// -----
conn.disconnect();
%>
```

The federated search JSP file

The sample code in Example 10-3 reads the query strings (InsuranceName, InsuranceDesc, InsuranceGroup and InsuranceNumber) from the request and executes a federated search query against the Information Integrator for Content server using these query attributes. After that, it loops through the result set and displays any available attributes of the found Content Manager content documents.

Copy and paste the sample JSP file as shown in Example 10-2 on page 222 into <WEB_APP_HOME>\jsp\FEDSearch.jsp.

Note:

- ▶ The *servertype* is *FED*.
- ▶ The federated search, by default, returns the latest version only.
- ▶ The example assumes that a search template has already been created in Information Integrator for Content with the four attributes InsuranceName, InsuranceDesc, InsuranceGroup, and InsuranceNumber.

Example 10-3 FEDSearch.jsp

```
<%@ page import="com.ibm.mm.beans.*, java.net.URL"%>
<%
// -----
// Read the posted parameters
// -----
String InsuranceName[] = request.getParameterValues("InsuranceName");
String InsuranceDesc[] = request.getParameterValues("InsuranceDesc");
String InsuranceGroup[] = request.getParameterValues("InsuranceGroup");
String InsuranceNumber[] = request.getParameterValues("InsuranceNumber");

// -----
// Define connection parameters of type FED
// -----
String servename = "ICMNLSDDB";
String servertype = CMBBaseConstant.CMB_DSTYPE_FED;
String userid = "icmadmin";
String password = "password";
String templateName= "InsuranceType";
```

```

// -----
// Connect
// -----
CMBConnection conn = new CMBConnection();
conn.setServerName(servername);
conn.setDsType(servertype);
conn.setUserid(userid);
conn.setPassword(password);
conn.connect();

// -----
// Get the Search Template from schema and configure it
// -----
CMBSchemaManagement schema = conn.getSchemaManagement();
CMBSearchTemplate searchTemplate = schema.getSearchTemplate(templateName);
searchTemplate.setMaxResults(100);
searchTemplate.setAsynchSearch(false);

// -----
// Set search criterions from the post parameters
// -----
searchTemplate.setSearchCriterion("InsuranceName", CMBBaseConstant.CMB_OP_LIKE,
InsuranceName);
searchTemplate.setSearchCriterion("InsuranceDesc", CMBBaseConstant.CMB_OP_LIKE,
InsuranceDesc);
searchTemplate.setSearchCriterion("InsuranceGroup",
CMBBaseConstant.CMB_OP_LIKE, InsuranceGroup);
searchTemplate.setSearchCriterion("InsuranceNumber",
CMBBaseConstant.CMB_OP_LIKE, InsuranceNumber);

// -----
// Run the federated query
// -----
searchTemplate.runQuery();

// -----
// Get the resultset and loop over it
// -----
CMBDataManagement dm = null;
CMBSearchResults result = new CMBSearchResults();
result.setConnection(conn);
result.newResults(searchTemplate.getResults());

%><TABLE border=1><%
for (int i = 0; i < result.getCount(); i++)
{
%><TR><%
    CMBItem item = result.getItem(i);

```

```

// -----
// Header Row
// -----
if (i == 0) {
%> <TR bgcolor=FFF000><%
    for (int y = 0; y < item.getAttrNames().length; y++)
    {
%>     <TD><%= item.getAttrName(y) %></TD><%
    }
%> </TR><%
    }
    dm = conn.getDataManagement();
    dm.setDataObject(item);
    URL url = dm.getContent(0).getDataURL();
    for (int y = 0; y < item.getAttrValues().length; y++)
    {
%>     <TD><a href="<%= url.toString() %>" target="Result"><b><%=
item.getAttrValue(y) %></b></a></TD><%
    }
%></TR><%
    }
%></TABLE><%

// -----
// Disconnect
// -----
conn.disconnect();
%>

```

Creating the necessary LWWCM objects

Once the JSP file is created, we need to create the necessary LWWCM objects to display the search options on the Web page and incorporate the created JSP file.

Creating a content template

Create a new content template called *NSE Template (FED Template)*. Use the component manager to add a new JSP component and call it *NSESearch (FEDSearch)*. This template will later be used to create a new content object representing the page which will be shown when the *NSE Search (FED Search)* site area is opened.

Creating the search entry form

This form represents the form to enter a new query. For NSE, it has only one input field named query. For the federated search, it has four input fields

representing the attributes defined in the corresponding Information Integrator for Content search template. When clicking the **Submit** button, the entered attribute/value pairs will be posted to /Intranet/Tools/NSE+Search (/Intranet/Tools/FED+Search) which is actually a call to the same site area and page again, since the search results will be shown right underneath the search entry fields.

For the *NSE search entry form*, create a new HTML component in the component library and name it *HTML - NSE Form*. Copy and paste the HTML code shown in Example 10-4 into the HTML component.

Example 10-4 HTML - NSE Form

```
<FORM METHOD=POST name=search action=<PathCmpnt
type="servlet"/>/Intranet/Tools/NSE+Search>
<TABLE><TR>
  <TD>Query:</TD> <TD><INPUT TYPE=Text value="%" size=56 name=query></TD>
</TR>
<TR>
  <TD><input type="submit" value="Execute NSE" name="submit"></TD><TD></TD>
</TR></TABLE>
</FORM>
```

For the *FED search entry form*, create a new HTML component in the component library and name it *HTML - FED Form*. Copy and paste the HTML code as shown in Example 10-5 into the HTML component.

Example 10-5 HTML - FED Form

```
<FORM METHOD=POST name=search action=<PathCmpnt
type="servlet"/>/Intranet/Tools/Federated+Search>
<TABLE>
<TR><TD>InsuranceName:</TD> <TD><INPUT TYPE=Text value="%" size=56
name=InsuranceName></TD></TR>
<TR><TD>InsuranceDesc:</TD> <TD><INPUT TYPE=Text value="%" size=56
name=InsuranceDesc></TD></TR>
<TR><TD>InsuranceGroup:</TD> <TD><INPUT TYPE=Text value="%" size=56
name=InsuranceGroup></TD></TR>
<TR><TD>InsuranceNumber:</TD> <TD><INPUT TYPE=Text value="%" size=56
name=InsuranceNumber></TD></TR>
<TR><TD><input type="submit" value="Execute FED Search"
name="submit"></TD><TD></TD></TR>
</TABLE>
</FORM>
```

Creating the page design

Create a new page design from the existing menu layout. To do this, open the menu layout and click **Save as**. Call the new page design *NSE Layout (FED Layout)*. In the appropriate section of the layout, first put the search entry form and then the JSP component onto the Web page.

Specifically, for NSE layout tags, add the following two lines:

```
<ApatrixLibCmpnt name="HTML - NSE Form"/><br>
<ApatrixCmpnt context="current" key="NSESearch" type="content"/>
```

For FED layout tags, add the following two lines:

```
<ApatrixLibCmpnt name="HTML - FED Form"/><br>
<ApatrixCmpnt context="current" key="FEDSearch" type="content"/>
```

Creating the content

Create a new content object called *NSE (FED)*. When creating the new content, use the *NSE Template (FED Template)* template created in “Creating a content template” on page 226 and place it under **Intranet -> Tools -> NSE (FED)**.

Tip: If the site area **Intranet -> Tools -> NSE (FED)** is not yet created, move the content once it is created (“Creating the site area” on page 229).

On the Contents tab of the new content window, define the values for the JSP component (see Figure 10-8 for the federated search content):

- ▶ JSP Path: /jsp/NSESearch.jsp (/jsp/FEDSearch.jsp)
- ▶ Error Message: Error executing Search (or something similar)

The screenshot shows a web application window with several tabs: ID, Properties, Contents (selected), Profile, Workflow, Security, and History. The 'Contents' tab is active, displaying a form for editing content. The form has three main sections: 'Display Title' with a text box containing 'Federated Search'; 'FedSearch' with a 'JSP Path' text box containing '/jsp/FEDSearch.jsp'; and 'Error Message' with a text box containing 'FEDSearch failed or no results!'. Each text box has a small upward and downward arrow icon on its right side.

Figure 10-8 The content object including the JSP component

On the Workflow tab, select **Site Administrators Express Workflow** as the workflow.

Creating the site area

Create new site area under **Intranet -> Tools** called *NSE Search (FED Search)*.

On the Properties tab, define the following (see Figure 10-9):

- ▶ Default Content: select **NSE (FED)**.
- ▶ Content Template: select **NSE Template (FED Template)**.
- ▶ Page Design: select **NSE Layout (FED Layout)**.

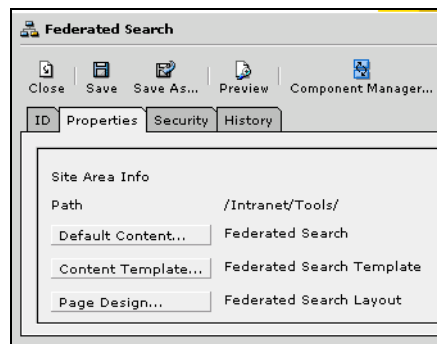


Figure 10-9 Site area properties

Whenever this site area is opened, it will show the content called NSE (FED) in the context of the page design called NSE Layout (FED Layout). The result is that the search entry form and the JSP component are rendered into the page.

Testing

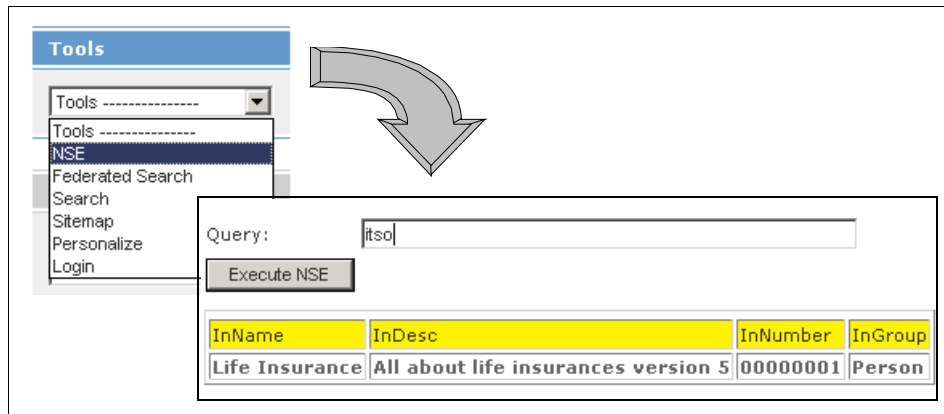
Open your Web browser and enter the URL to the intranet home page:

`http://[SERVER]:[PORT]/[CONTEXT]/connect/intranet`

On the right-hand side, open the tools drop-down box and select **NSE**. Now you can enter queries and click **Execute NSE (Execute FED)** to start executing searches. The result should look similar to the following screen shots (Figure 10-10 on page 230 for NSE and Figure 10-11 on page 230 for the federated search). The results are shown in a table as links. If you click the link, the attached Content Manager content will be opened.

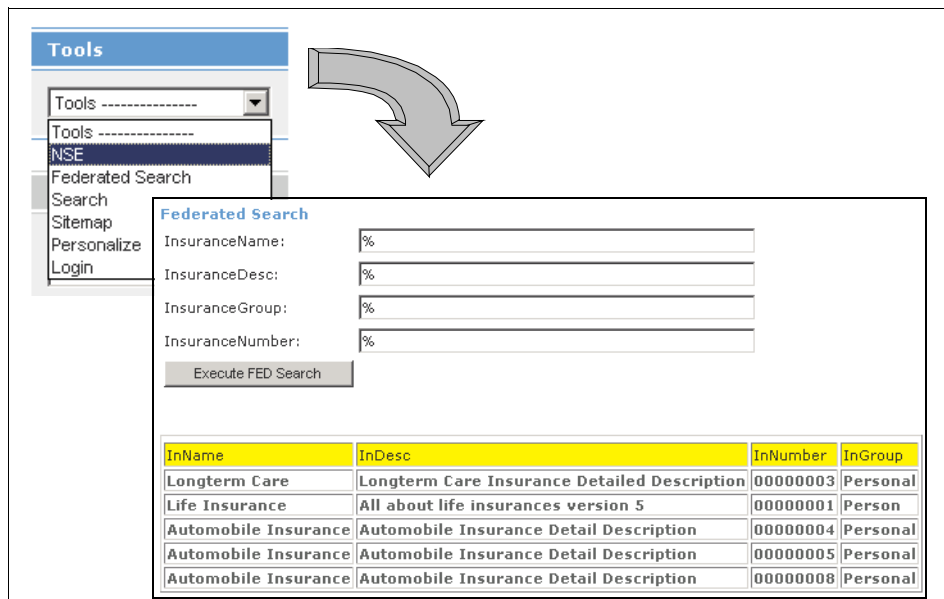
As mentioned earlier, with some extra effort, you can combine the NSE search and the federated search into a seamless search interface for users. Users should not need to decide whether they need NSE or a federated search. You should design your search interface to be easy to understand and easy to

operate. What we show here is for demonstration purposes only. We do not recommend that you actually implement this with the exact same wording.



InName	InDesc	InNumber	InGroup
Life Insurance	All about life insurances version 5	00000001	Person

Figure 10-10 NSE and search result Web page



Federated Search

InsuranceName: %

InsuranceDesc: %

InsuranceGroup: %

InsuranceNumber: %

Execute FED Search

InName	InDesc	InNumber	InGroup
Longterm Care	Longterm Care Insurance Detailed Description	00000003	Personal
Life Insurance	All about life insurances version 5	00000001	Person
Automobile Insurance	Automobile Insurance Detail Description	00000004	Personal
Automobile Insurance	Automobile Insurance Detail Description	00000005	Personal
Automobile Insurance	Automobile Insurance Detail Description	00000008	Personal

Figure 10-11 Federated search and search result Web page

10.2.2 IFrame

If you already have a Web application that provides searching capabilities, you might be able to reuse this Web application using the iFrame technology within your Web page if the Web application complies with some basic rules.

Note: The iFrame element creates an in-line frame that contains another Web page.

The basic rules apply to all Web applications you want to display in iFrames and are not directly related to the search integration into LWWCM:

- ▶ The Web application needs to support iFrames in general. Links within the application should either open a new browser window or reference the URL to be opened relative to the current path. In other words, a link must be programmed in such a way that it does not lead to the parent window accidentally.
- ▶ The user credentials to access the back-end system must be accessible to a cross-browser session in order to prevent continuous user/password requests.

In the following example, we have taken the Content Manager eClient Web application as the source for the iFrame. Figure 10-12 on page 232 shows the resulting Web page.

Restriction: Although we use eClient as an example Web application to be the source for the iFrame, the eClient Web application *does not* comply with the basic rules required to integrate with iFrame. The reasons why are as follows:

- ▶ Some buttons affect the parent window and therefore destroy your LWWCM Web page. In order to prevent this, select the **Open in new window** checkbox prior to performing a search (see Figure 10-12).
- ▶ The iFrame source URL taken requires the user to be successfully authenticated against the eClient Web application in the current session.

In order to achieve the integration of an external Web application in your LWWCM Web page, do the following:

1. Create a new HTML library component holding the following HTML:

```
<iframe  
src="http://demo01:9080/eClient82/IDMBasicSearch.jsp?ServerName%3DDEM02DB%2  
6serverType%3DICM%26userid%3Dicmadmin%26language%3Den&entityName=InsuranceT  
ype" width="100%" height="350">  
</iframe>
```

2. Reference the new HTML library component in a page design of your choice using the following tag:

```
<ApatrixLibCmpnt name="<Name of the HTML library component>" />
```

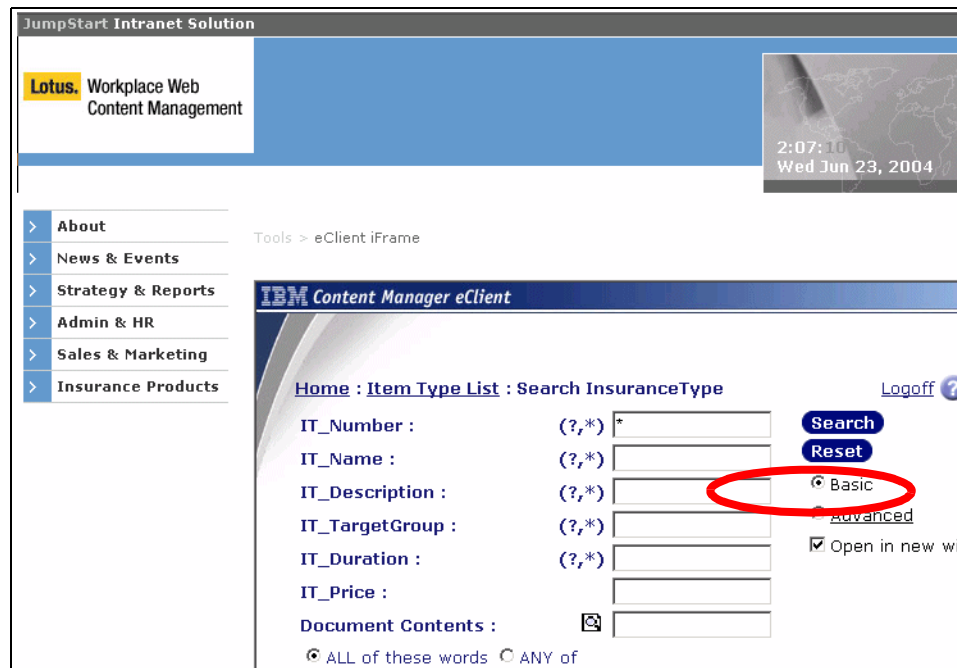


Figure 10-12 eClient integrated through an iFrame

10.2.3 LWWCM internal search

The search module of LWWCM is limited to objects and keywords stored in the LWWCM environment only. It is independent from the data repository type you use to store your LWWCM objects. This means that no matter which repository type (file system, DB2, or Content Manager) you use, search results with identical queries will be the same. Security on the LWWCM object level is taken into account. A user will therefore only receive hits on content he or she is allowed to see.

Searching external content (for example, content in Content Manager) is not supported unless you include the external content into your Web page using the *Federated Content Component*. Only text content which is included on the Web page is incorporated in the search index. To make sure that all the attributes and the actual content of the federated content are searchable, you have to render them all directly onto your Web page using the Federated Content Component

tag or the Federated Content Library Component tag. Refer to 7.3, “Federated content component” on page 123 for more information on how to use these tags.

Restriction: Only content of MIME type text/* can be rendered into a page; therefore, content of other MIME types, such as Microsoft Word documents, cannot be searched using the LWWCM internal search engine.

To make all Content Manager objects of the item type InsuranceType searchable regarding the four attributes (Name, Description, Number and Group) and the attached Content Manager content of MIME type text/* searchable, you would have to add the tags given in Example 10-6 to your page design and create at least one LWWCM content page per insurance type, including a federated content component referencing it.

Example 10-6 Tags in page design to enable search on attributes and content

```
<b>Name:</b> <FederatedCmpnt AttributeName="InName" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Description:</b> <FederatedCmpnt AttributeName="InDesc" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Number:</b> <FederatedCmpnt AttributeName="InNumber" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Group:</b> <FederatedCmpnt AttributeName="InGroup" context="current"
key="FedCmpnt" Scope="attribute" type="content"/><br>

<b>Content:</b> <br> <FederatedCmpnt context="current" key="FedCmpnt"
Scope="content" type="content"/>
```

The resulting Web page would be similar to Figure 10-12 on page 232. Queries on any word within this Web page would render this page in their result set. For example, if you omitted the tag which renders the content (Scope="content"), only queries on any word included in the attributes would have this page in their result set.

Tip: LWWCM search settings such as hit index refresh, directory to store the search index in, or an ignore list can be configured in:

[LWWCM_HOME]\config\aptrixsearch.properties

10.3 Other search approaches

There are other search approaches available. We discuss two of them briefly:

- ▶ Portal search engine (PSE)
- ▶ Domino Extended Search (DES)

10.3.1 Portal search engine (PSE)

The portal search engine (PSE) comes with the WebSphere Portal Server (WPS). It is a search application that crawls Web sites and returns a complete set of documents/pages or a subset based on filter criteria. Optionally, it can categorize and summarize the content and store the data according to a selected taxonomy, which is either user-defined or a pre-defined system taxonomy. Another option available with PSE is a simple workflow that provides document approval and metadata update features.

The created index is a full-text index. While indexing, PSE applies document filters of more than 250 document formats. The technology used is Stellent's Outside In technology.

Restriction: Currently, the PSE can only search pages where anonymous access is granted.

Administration and actual use of the search functionality are implemented through a set of portlets.

Tip: For more information, go to the WebSphere Portal Info Center located at:

<http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html>

The PSE can be set up to crawl a Web page based on LWWCM. In our insurance company example, the site to crawl would be:

`http://[SERVER]:[PORT]/[CONTEXT]/connect/intranet`

The crawler will crawl the given URL and as many levels of subsites as defined. If your site contains a link to a content in Content Manager, the crawler will also fetch this linked document and incorporate the content in its full-text index if the file type is supported.

10.3.2 Domino Extended Search (DES)

DES is not a search engine in and of itself. It does not create its own search index. It leverages other search engines and works as a broker on your behalf. It

returns aggregated and personalized results from multiple sources as a single, ranked list of items that matches your search criteria. It enables you to search structured and unstructured data through a single point of access and it honors the security models of the back-end systems.

The connections to the source search engines are done through links. Out of the box, the following links are provided:

- ▶ Notes databases
- ▶ Domino.Doc®
- ▶ Quickplace
- ▶ Domino R5 Domain Index
- ▶ Domino Full Text Index
- ▶ File System
- ▶ IBM DB2
- ▶ Oracle
- ▶ Sybase
- ▶ MS SQL Server
- ▶ MS Index Server & Indexing Service
- ▶ LDAP
- ▶ Web search engines

DES has a set of comprehensive Java APIs which can be used to extend it with new links to additional search engines accessible through Java APIs. This feature could be used to develop a new Content Manager/Information Integrator for Content link using the federated search or NSE API.

The search front end to DES is Web-based and fully customizable to fit your needs. It could, therefore, be integrated into your LWWCM Web page without a major effort. There are also portlets available for WebSphere Portal which can be used out of the box.

Tip: For detailed information about DES, see the following Web pages:

- ▶ DES home page:
<http://www.lotus.com/products/des.nsf/wdocs/home>
- ▶ WebSphere Portal and Lotus Workplace catalog:
<http://catalog.lotus.com/wps/portal/portalworkplace>



Security

In this chapter, we describe how LWWCM and Content Manager security work, how they work together, and the access level in LWWCM.

We discuss how users have access to objects on the live Web site and how they have access to the objects within the LWWCM user interface.

We also describe the security background working with:

- ▶ Content Manager components
- ▶ Connect tag
- ▶ JSP component

11.1 LWWCM security

Users who need to have access to the LWWCM user interface must be defined in the *Security & Workflow* section of the LWWCM user interface. Users who browse the Web sites created from LWWCM do not need to access the LWWCM user interface. Users who only need to browse, not produce or manage the content published by LWWCM, do not need to be defined in the Security & Workflow section of the LWWCM user interface.

All objects created in LWWCM have a security setting. In the object definition window, click the **Security** tab to get to the setting panel. From this panel, you can define user or group access and access levels for the LWWCM object. There are different access levels which can be set to access LWWCM objects. Object security settings must be defined in the Security tab window of an object while the workflow security can be defined for workflow and stage objects.

11.1.1 Access levels and security sections

Object security is set in the Security tab of the object definition window. There are four access levels you can set for an LWWCM object. Figure 11-1 shows the security panel for an LWWCM object, site.

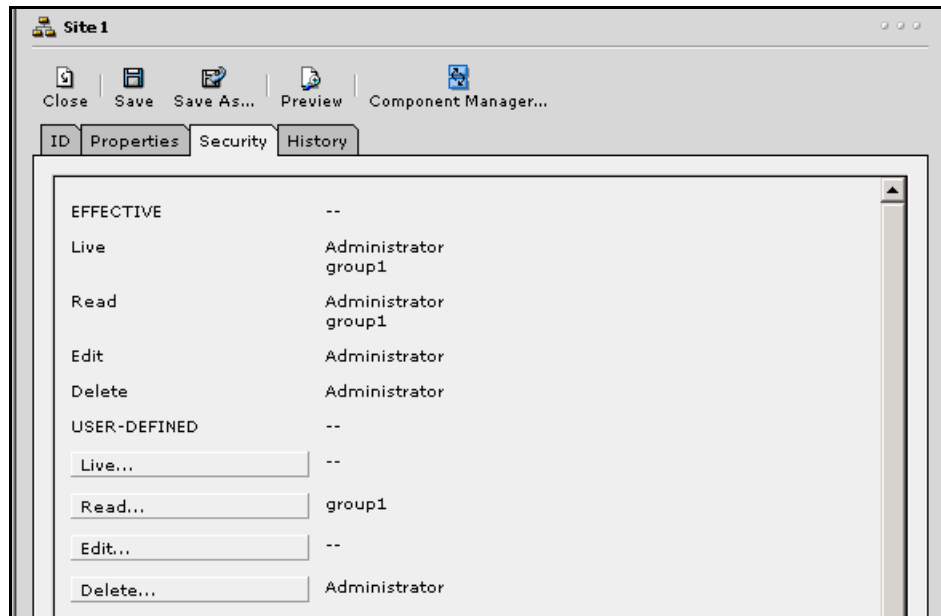


Figure 11-1 LWWCM object security setting

Four access levels include:

► **Live**

This level allows users or groups to view the object in the *Live* Web site. If the object is defined in a workflow, then the object is on the live view state when it is published.

► **Read**

This level allows users or groups to:

- View the object in the *Live* Web site.
- View the object in the LWWCM user interface.

► **Edit**

This level allows users or groups to:

- View the object in the *Live* Web site.
- View the object in the LWWCM user interface.
- Edit the object in the LWWCM user interface.

► **Delete**

This level allows users or groups to:

- View the object in the *Live* Web site.
- View the object in the LWWCM user interface.
- Edit the object in LWWCM user interface.
- Delete the object in the LWWCM user interface.

Tip: If you set users or groups Delete access level to an object, they automatically get Edit, Read and Live access rights to this object.

Workflow security is defined on the Properties tab window of the workflow stage objects. You can define different access levels: Live, Read, Edit, Delete , and, additionally, workflow-specific rights.

► **Approve**

This level allows users or groups to:

- View the object in the *Live* Web site.
- View the object in the LWWCM user interface.
- Approve an object within a workflow.

Figure 11-2 on page 240 shows sample workflow buttons. Users with proper approve rights can move objects from one workflow stage to the next.

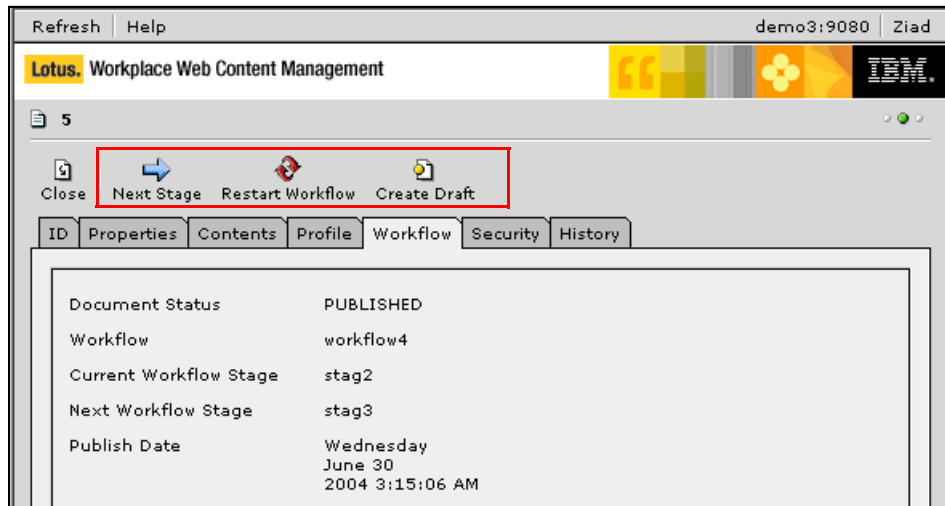


Figure 11-2 Workflow buttons

When you create a new object, you automatically have all the access rights to the object (Live, Read, Edit, and Delete). If the new object has become part of a workflow, the security setting of the object is applied only to the first workflow stage. As the object goes through the subsequent workflow stages, the workflow security of the particular workflow stage will be added to the object security and dictate the security setting of the object.

On the security setting window of an object, the following security sections are available:

- ▶ **Effective:** Specify the effective security for the object. It is the result of the combined User-Defined or Workflow + System-Defined access levels.
- ▶ **User-Defined or Workflow:** If the object is not part of a workflow, user defined access can be set. If the object is part of a workflow, then the security tab of this object will show the workflow security setting and not the user-defined setting. The current workflow stage of the object within a workflow determines the access level for this object.
- ▶ **System-Defined:** Specify the access level set by the administrators. All members of the administrators group can edit and set the system defined security section.

11.1.2 Default users and groups

Before you assign users and groups to different access levels within different security sections, you must define the users and groups in the **Security &**

Workflow -> Users & Groups section of the LWWCM user interface. During LWWCM installation, two default users and five default groups are created.

Default users

The two default users created during LWWCM installation are:

- ▶ **Administrator:** This is the default administrator of the LWWCM user interface. The administrator is a member of the *Administrators* group and has access to *every object* of the LWWCM user interface. The default password is password. We recommend changing it immediately after the LWWCM installation.
- ▶ **Anonymous:** Anonymous users are used for public access to the LWWCM live Web sites. The anonymous users do not need a password. If only the Anonymous user has access to an object in the LWWCM user interface, then users who have already logged on will be unable to access the object. Because of this, we recommend that the Anonymous user:
 - Only be given *Live* access to objects.
 - Not be added to the user interface Access group. This prevents the Anonymous user logging into the LWWCM user interface.

Default groups

Five default groups, defined during the LWWCM installation, enable their members different access levels to the LWWCM user interface and to live Web sites. The default groups are:

- ▶ **Administrators:** The members of this group have access to all elements in the LWWCM user interface including the LWWCM application files (the files that build the interface and begin with “AptribUI”). By default, the user, Administrator, is a member of this group.
- ▶ **UI Access Group:** The members of this group have access to the LWWCM user interface. Users who create, design, or administer the Web site must be added to this group. Users who only need to access the live Web site, that is the published Web site, do not need to be added to this group. When a user create an object, the UI Access Group is automatically added to the System-Defined Read field by default.
- ▶ **[ALL Users]:** All users are members of this group *including the Anonymous users*.
- ▶ **Version Creators:** The members of this group have the ability to create versions for object and have access to the **All Version** and **Version** buttons (functions) in the user interface.
- ▶ **Restorers:** The members of this group can restore a version from the Version library. They have access to the *Restore* and *Restore as draft* buttons (functions) in the user interface.

11.1.3 How it works

In the previous sections, we introduced the four access levels (Live, Read, Edit, and Delete) of an object security, and the additional Approve access level in a workflow stage object. We cover the four security sections in the security setting of an object (Effective, User-Defined or Workflow, and System-Defined). We describe two default users (Administrator and Anonymous) and five default groups (Administrators, UI Access Group, All Users, Version Creators, and Restorers) created during LWWCM installation. In this section, we tie all these security concepts together and go into detail as to how the LWWCM security works.

LWWCM security uses the following rules:

- ▶ A user needs, at a minimum, Live access level to the objects that the user needs to see on a live Web site. This includes all published content on the Web site.
- ▶ When an author wants to create a new content, the author needs to select a template, a site or site area where the content will be linked to and a workflow that the newly created content has to go through. The author must have access to these three objects (template, site or site area, workflow) in order to be able to select them. This means that the author needs, at a minimum, Read access level in the security setting (tab) of these three objects.
- ▶ When an object participates in a Workflow, then the security level depends on the workflow stage where the object is. In addition, the author of the object will be added to the workflow-defined access in the Delete access level for the first stage. When the object moves to the next stage, the security of the object will be pulled from the workflow security of that stage (which is defined in the Properties tab of that workflow stage object).

Note: If you define a default author in the Content Template tab of a template, then this default author will be added to the Delete access level in the object created using this template for the *first* workflow stage.

Here is an example.

The Administrator logs on and performs the following task:

1. Create three users: user1, user2, user99.
2. Create a group: group1 and add user99 to the group. The member of this group will be the author for new content.
3. Create two workflow stages: stage1 and stage2. In the Properties tab:
 - For stage1, under the Workflow Security setting, set the following:
 - Approve: group1

- For stage2, under the Workflow Security setting, set the following:
 - Read: User2
 - Delete: Administrator
 - Approve: group1

Note: Do not confuse a workflow security setting that you define in the Properties tab of workflow stage with the security setting that you define in the Security tab of workflow stage. *The workflow security in the Properties tab* is for all objects that are part of a workflow and enter this stage. *The Security tab* is for security settings of the workflow stage object itself. It defines who can delete, edit, or read this object.

4. Create a workflow object: workflow4. In the Properties tab, enter stage1, stage2 in the Workflow Stages field. In the Security tab, add group1 to the Read field in the User-Defined security (so an author which is a member of group1 will be able to select this workflow when the author creates a new document).
5. Create a Page Design: pd1.
6. Create a Content Template: ct1. Add group1 to the Read field in the User-Defined security of ct1 (so the members of group1 can select this template when creating a new document). To demonstrate how the author will be added to the Workflow Security's Delete access level in the first Stage of the workflow, we select **user1** in the Authors field in the Content Template tab. See Figure 11-3 for the settings.

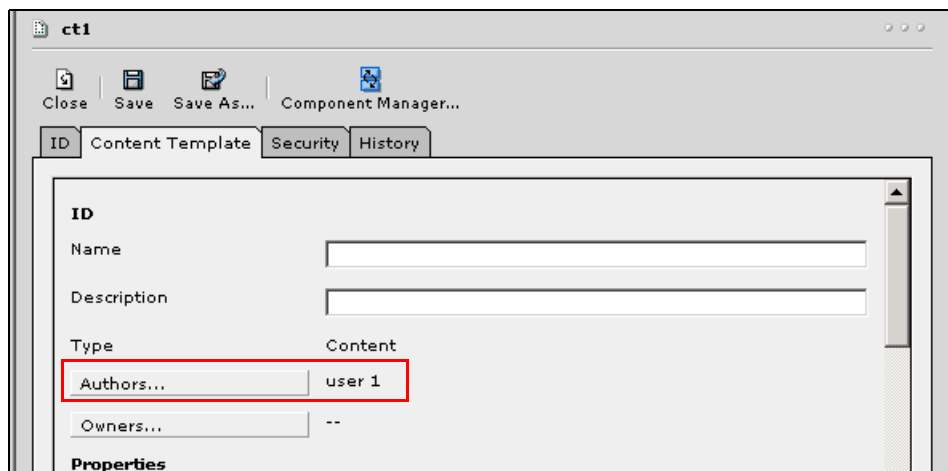


Figure 11-3 Default author for content created using this template

7. Create the site and site area (site1) and add group1 to the Read field in the User-Defined security (so the members of group1 will be able to select this site area when creating a new document). In the Properties tab, add the Page Design pd1 and Content Template ct1. See Figure 11-4 for the settings.

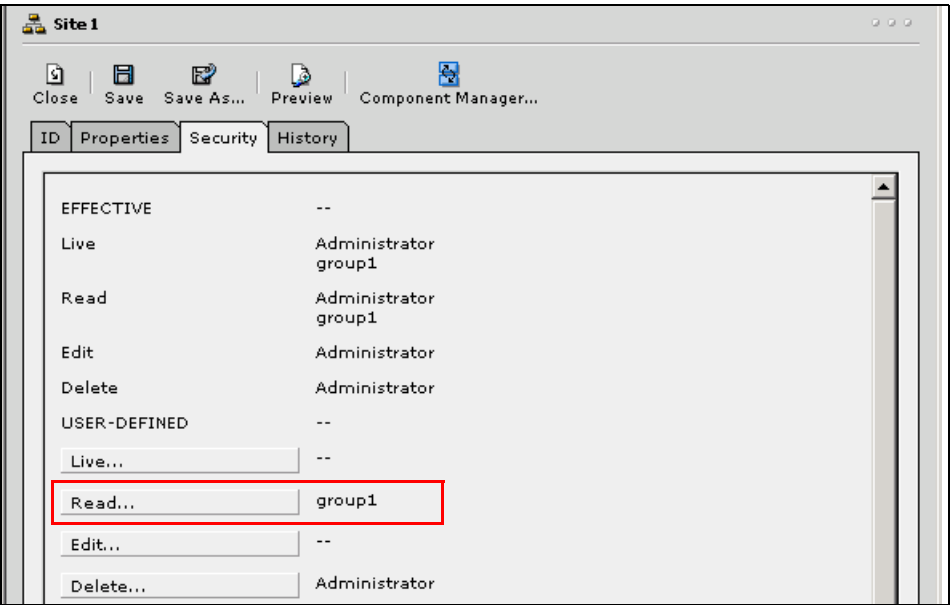


Figure 11-4 Add group1 so that the members of this group can see this site

Table 11-1 summarizes the administrator's setup.

Table 11-1 Example setup

Object type	Name	Setting
User	user1, user2, user99	
User Group	group1	Contains user99
Workflow Stage	stage1	Workflow Security, Approve: group1
Workflow Stage	stage2	Workflow Security, Approve: group1 Read: user2, Delete: administrator
Workflow	workflow4	Contains stage2 and stage2
Page Design	pd1	
Content Template	ct1	User-Defined, Read: group1, Author: user1

Object type	Name	Setting
Site Area	site1	User-Defined, Read: group1 Page Design: pd1, Content Template: ct1

Once the administrator finishes the tasks, the user, user99, can log in and create new content. The user99 performs the following tasks:

1. Select **Home -> Content Management -> Content Library** and click **New** for a new content.
2. Select **ct1** for a Content Template, then select a site area.
3. In the ID tab, enter a name for the document (content test 1). Note the author of this new content is user99 *and* user1 which inherits from the Content Template ct1. See Figure 11-5.

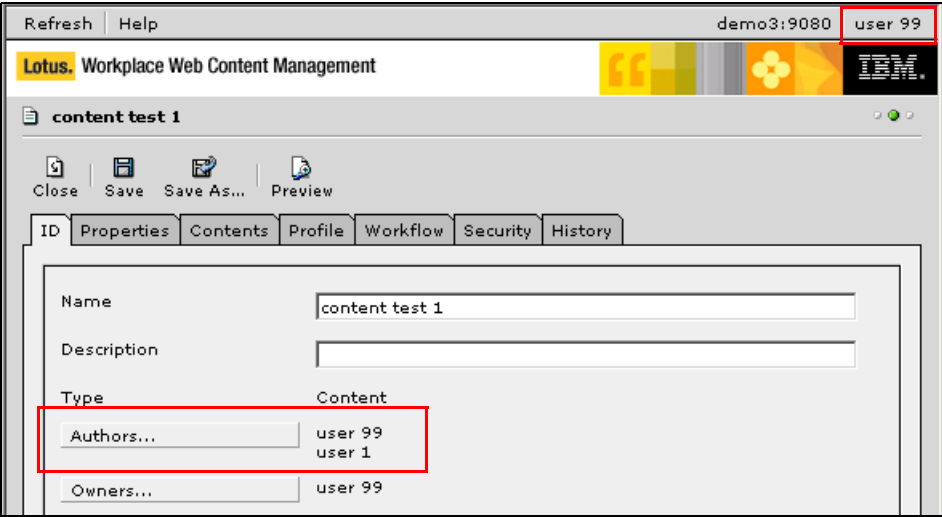


Figure 11-5 Authors field inherit from Content Template's Author field

4. Create some content in the Content tab and profile in the Profile tab if necessary.
5. In the Workflow tab, select **workflow4** as the workflow.
6. Save the object.
7. Go to the Security tab. See Figure 11-6 on page 246 for the following:
 - When the object is saved and goes to the first stage of the workflow, it inherits the security from the workflow security setting defined in the Properties tab of the workflow stage stage1. (Note the Approve always has Read access level.)

- The Authors are added to the Delete field: user99 and user 1 from the Content Template tab Author field of the Content Template ct1.

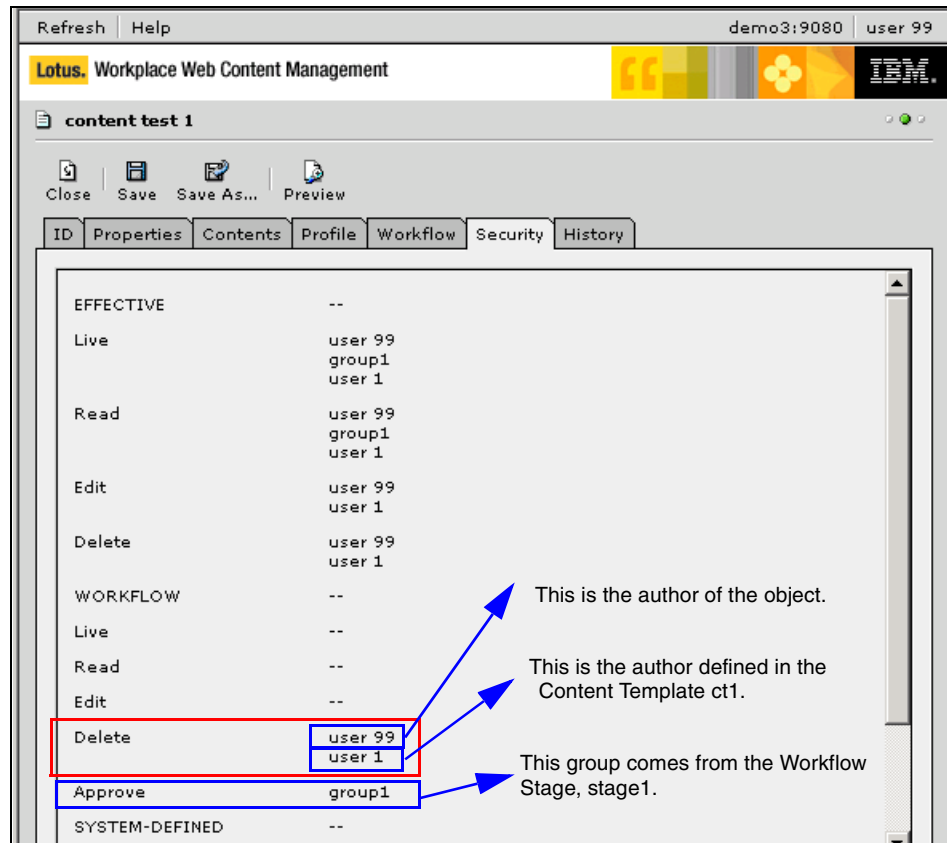


Figure 11-6 Security is from the workflow stage and the author is added

8. Go to the Properties tab of the Workflow Stage, stage1. See Figure 11-7 on page 247.

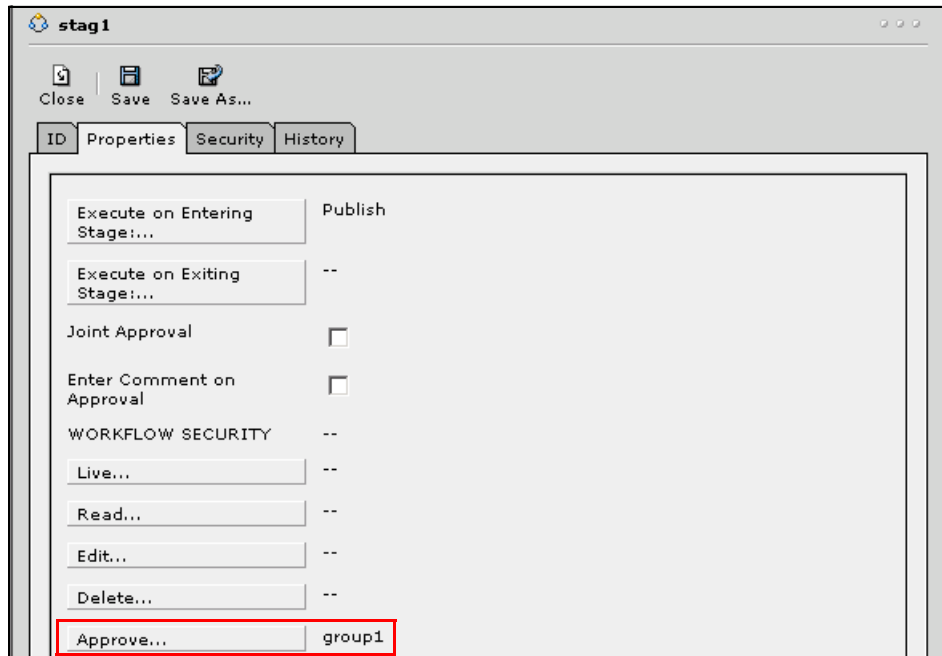


Figure 11-7 Workflow Security setting in stage1

9. Open the Document (content test 1) in read mode and move it to the next Workflow Stage (stage2) by clicking **Next Stage**. Note that the setting in the Security tab has changed to the setting defined in the Properties tab of the Actual Workflow Stage, stage2. See Figure 11-8 on page 248 for the Workflow security settings in the Properties tab of stage2 we defined during the creation of the Workflow Stage as the following:
 - Read: user 2
 - Delete: Administrator
 - Approve: group1

If you take a look in the Security tab of the document, content test1 (also see Figure 11-8 on page 248), we see that the workflow security setting has changed. Once the document moves to stage2, it inherits the security setting in the Properties tab of stage2.

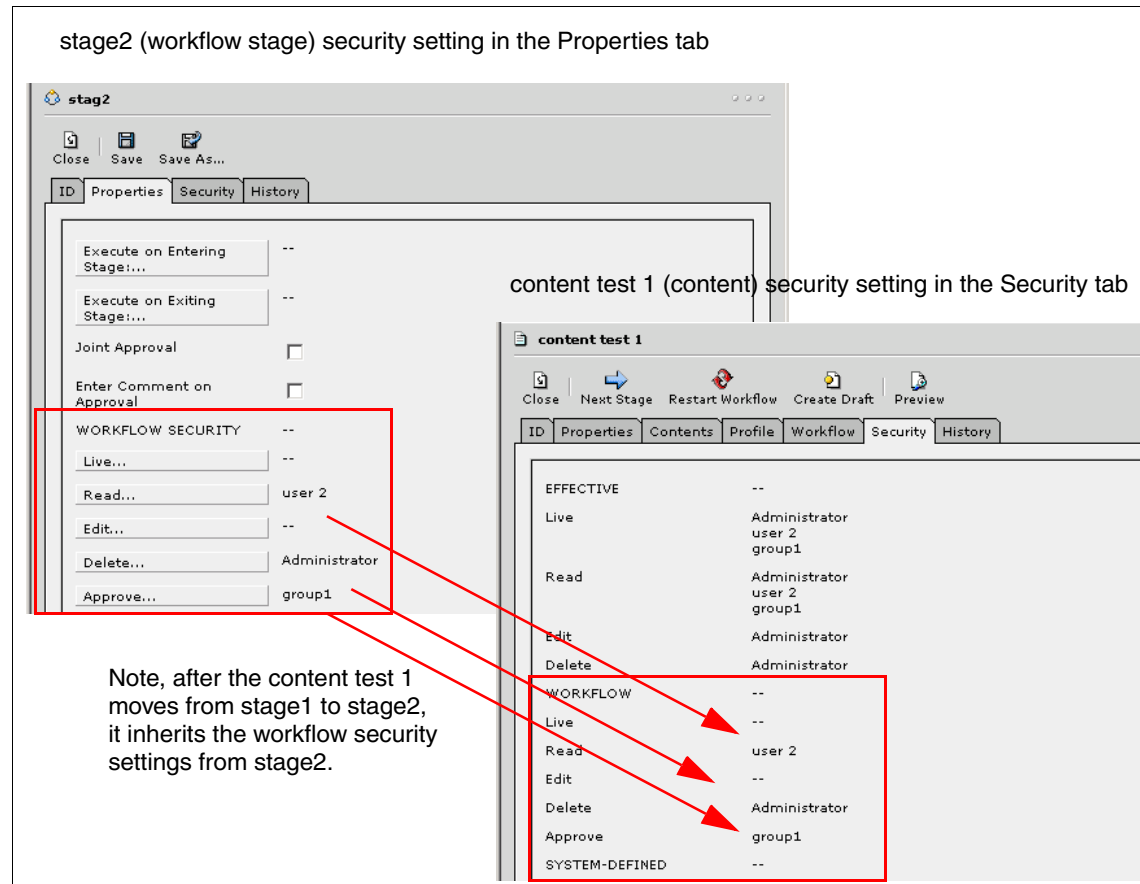


Figure 11-8 Workflow security setting for content test 1 after moving to stage2

11.2 Content Manager security

The Content Manager security is defined in the Content Manager Library Server using the Content Manager System Administrator Client. To set up the security, you have to:

1. Create a privilege.
2. Build a privilege group.
3. Defined a privilege set.
4. Generate an access control list.
5. Define the user and groups.

In this section, we briefly introduce some of the concepts involved. To learn more about Content Manager security, refer to *Content Manager Implementation and Migration Cookbook*, SG24-7051.

11.2.1 Authorization

Privileges, privilege sets and access control lists are used to authorize a user or a group to have certain access rights, such as write, read, delete, and search, in Content Manager:

- ▶ **Privilege:** Permission to perform an action on an object or item. There are system-defined privileges and user-defined privileges. An example of system-defined privileges is ItemDelete, which allows a user or a group to delete an item.
- ▶ **Privilege group:** Represents a group of privileges that facilitates the creation of a privilege set.
- ▶ **Privilege set:** Defines the different privileges that you can give to users or include in an Access Control List (ACL). Users cannot have privileges directly. They can have privilege sets which represent the maximum access that they can have.
- ▶ **ACL:** Represents a list consisting of one or more users or groups and their associated privilege sets. You set ACL for the data objects such as item types, work nodes, processes and worklists. Work nodes, processes, and work lists are used in the document routing process.

For example, you can prevent a user accessing a specific document routing process by giving this process an ACL that contains the user's name associated with a privilege set of NoPrivs which forbids the user to access the process.

11.2.2 Authentication

Users and groups are used to authenticate a user to the Content Manager system:

- ▶ **User:** You define a user to the system and assign a privilege set. You can also use LDAP to import users to the Content Manager system.
- ▶ **Group:** A user can belong to one or to many groups. Groups are used to *conveniently organize users*. After assigning a group to a privilege set, any change in the members of the group *does not* affect the associated privilege set. ICMPUBLIC is a special group which contains all users. This group can be enabled using the Library Server configuration panel.

11.2.3 How it works

When a user tries to perform an action (such as search and retrieve) on an item, the Library Server uses the following precedence rules to check that the user has the right to perform that particular action on the item:

1. ItemSuperAccess: If the user has ItemSuperAccess privilege, the Library Server bypasses the ACL authorization and enables the user to continue.
2. ICMPUBLIC group: If ICMPUBLIC group is enabled in the Library Server configuration and the ACL of the item contains this group, then the Library Server checks if the privilege set assigned to ICMPUBLIC allows this particular action. If so, permission is granted, and the user can continue.
3. User: If the ICMPUBLIC is not contained in the ACL, the Library Server checks to see if the user is in the ACL. If so, the Library Server checks to see if the privilege sets assigned to the user allow this particular action. If so, it grants the access.
4. Group: If the user is in a group and this group is in the ACL, the Library Server checks to see if the privilege sets assigned to the *user* allow this privilege. If so, it grants the action and the user can continue.
5. If none of the above applies, the user cannot perform the specific action on this item.

Figure 11-9 on page 251 provides the flow chart for this decision-making process.

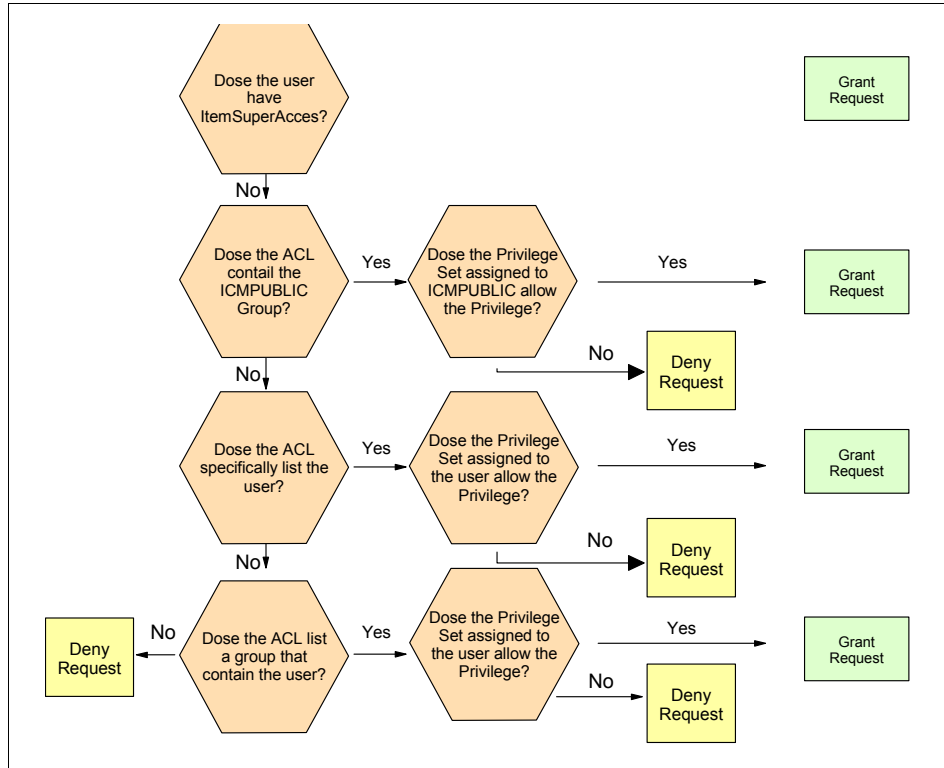


Figure 11-9 ACL security process flow chart

11.3 How security types work together

In the following section, we describe the security control for an integrated LWWCM and Content Manager solution and how it works.

11.3.1 Security in general

When a user logs in to an LWWCM user interface, the user must be defined in the LWWCM user interface with a valid user name and password. When a user creates an object, assuming LWWCM uses Content Manager as the back-end content repository, then LWWCM uses access information defined in the `aptrixjpe.properties` file to access Content Manager. The user does not need to know where and how the data is stored nor does the user need any special access to the Content Manager.

In the `aptrixje.properties` file, the Content Manager server name, user name, and password are set as follows:

```
# CM persistency settings
# CM Server information
cm.server=ICMNLSDb
cm.username=icmadmin
cm.password=password
# Unique identifier for the site
cm.jpeServerIdentifier=demo2
```

For more information on how to set this configuration, see 6.1.3, “Modifying LWWCM’s `aptrixje.properties` file” on page 94. This access information will be used by LWWCM to access Content Manager Library Server. If the information is not set, then the user needs to have the right access level to the Content Manager system.

11.3.2 Security with connect tags

Connect tags use a different security setting. You have to define the Content Manager server name, user name, and password in the connect tag. This enables you to access different Content Manager systems using different connect tags. The Content Manager server must also be defined in the `cmbicmsrvs.ini` file on the host where LWWCM and WebSphere Application Server run.

An example of the connect tag setting is as follows:

```
<connect MOD="TEMPLATE" SRV="CM" SERVER="icmnlbdb" USER="icmadmin"
PWD="password" SEARCH="/InsuranceType"
TEMPLATE="http://localhost:9080/ilwwcm/templates/template1.html"></connect>
```

For more information see 7.4.3, “Putting it all together” on page 148

11.3.3 Security with federated content component

The federated content component is used to store a reference to the Content Manager document that you use in LWWCM Web pages to show the document or attributes. When using the federated content component, the security information used to access Content Manager will be taken as the following:

- The Content Manager server has to be defined in the `connect.cfg` file in the LWWCM directory. See the following as an example:

```
<FederatedConnector>
  <DefaultServerAlias value = "ICMNLSDb" />
  <DefaultUsername value = "icmadmin" />
  <DefaultPassword value = "password" />
```



```
<MaxResultsReturned value = 100 />
</FederatedConnector>
```

- The user name and password used to access Content Manager is taken in the following order:
 - a. If the user name and password are defined in the federated content component (see Figure 11-10), they override authentication information defined in the connect.cfg tag.

Figure 11-10 Federated content component setting has the first priority

- b. If the user name and password are not specified in the federated content component object, they will be taken from the connector section of the connect.cfg file.
- c. If the user name and password are not defined in the connect.cfg file, then the user name and password of the currently logged-on user viewing the site will be used.

For more information, refer to 7.3, “Federated content component” on page 123.

11.3.4 Security with the JSP component

Using JSP components, you can access Content Manager data using Content Manager and Information Integrator for Content Java APIs. In the JSP component, you define the JSP path where the JSP file is located. The security information needed to access Content Manager data is defined in the JSP file. The following is an example of a JSP file accessing a Content Manager system:

```
// -----
// Setup the CM Connection
// -----
String servername = "ICMNLSDDB";
String servertype = CMBBaseConstant.CMB_DSTYPE_ICM;
String userid = "icmadmin";
String password = "password";
String query = "/InsuranceType[@VERSIONID = latest-version(.)]";

CMBConnection conn = new CMBConnection();
conn.setServerName(servername);
conn.setDsType(servertype);
conn.setUserid(userid);
conn.setPassword(password);
conn.connect();
```

For more information, refer to 7.5, “JSP component” on page 151.



VideoCharger integration

In this chapter, we describe how you can integrate IBM DB2 Content Manager VideoCharger (VideoCharger) to the LWWCM Web site.

Specifically, we address the following topics:

- ▶ VideoCharger and how it works
- ▶ Integration of VideoCharger and Content Manager
- ▶ Integrating videos stored in Content Manager to the LWWCM Web site

12.1 VideoCharger and how it works

IBM DB2 Content Manager VideoCharger (VideoCharger) is a server designed to handle large streaming video files. VideoCharger can deliver video and audio files in real time (this is known as *streaming*) to clients over a network. Video (audio) streaming enables client users to watch videos without having to wait for the entire media file to be downloaded. Media files can have a size of 100 MB and more. VideoCharger can also manage the storage of large media files. For the media formats that VideoCharger version 8 server supports, see 2.2.6, “IBM VideoCharger (bundled with Content Manager)” on page 20.

The IBM DB2 Content Manager VideoCharger player (VideoCharger player) can be used from a Web browser as a client to play video(s) that is streamed from a VideoCharger server. The VideoCharger server first will send a metadata file to the client and then stream the video or general media object to the player.

In our case study, we install the VideoCharger server and player, version 8.2. When you install the VideoCharger and VideoCharger player, as shown in Figure 12-1 on page 257, be aware of the following installation requirements:

- ▶ *First install the License Use Management (LUM) software.* This is necessary because it is used to manage the amount of concurrent streams delivered by a VideoCharger server. If it is not installed, you will get error ICM9799, as shown in Figure 12-8 on page 264, when you try to import any video from the Content Manager Windows Client.
- ▶ Before installing the VideoCharger player, be sure to close all Web browser windows; otherwise, the VideoCharger Web browser plug-in cannot be installed on your default browser window properly.
- ▶ For a VideoCharger V8.2 AIX installation, the prerequisite for AIX is V5.1 or higher.
- ▶ Refer to the VideoCharger manual for details on how to configure the IBM HTTP Server for use with the VideoCharger.

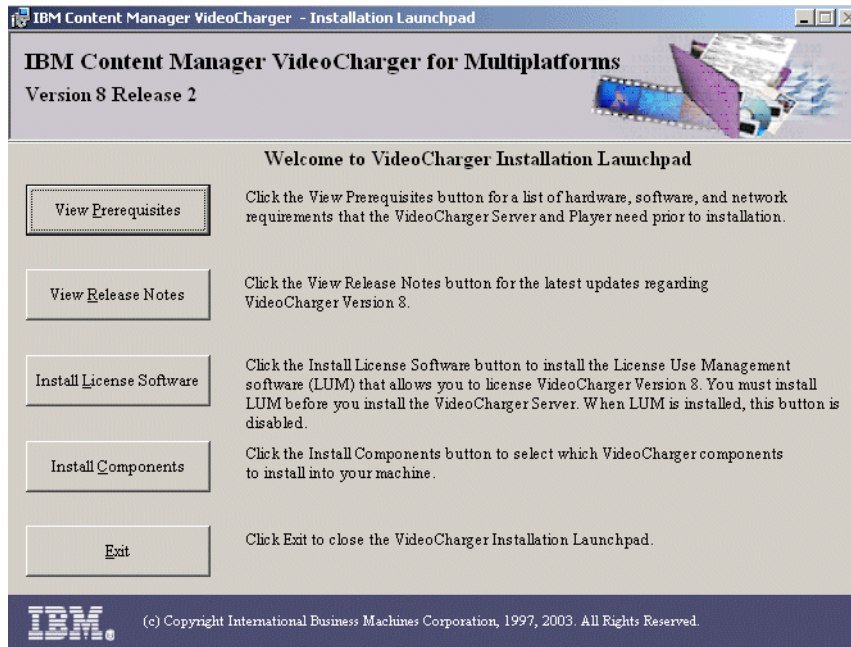


Figure 12-1 VideoCharger installation

After the installation, also consult the VideoCharger online documentation. Make the following changes:

1. Open a Web browser and go to http://localhost/vs_admin/ (see Figure 12-2 on page 258).
2. Recalibrate all network interfaces.
3. Change the network threshold to 80%.
4. Select **Update**.

This setup enables the VideoCharger server to use up to 80% of the network bandwidth to stream the videos to the used client. You may change it to another threshold value to fit your business environment.

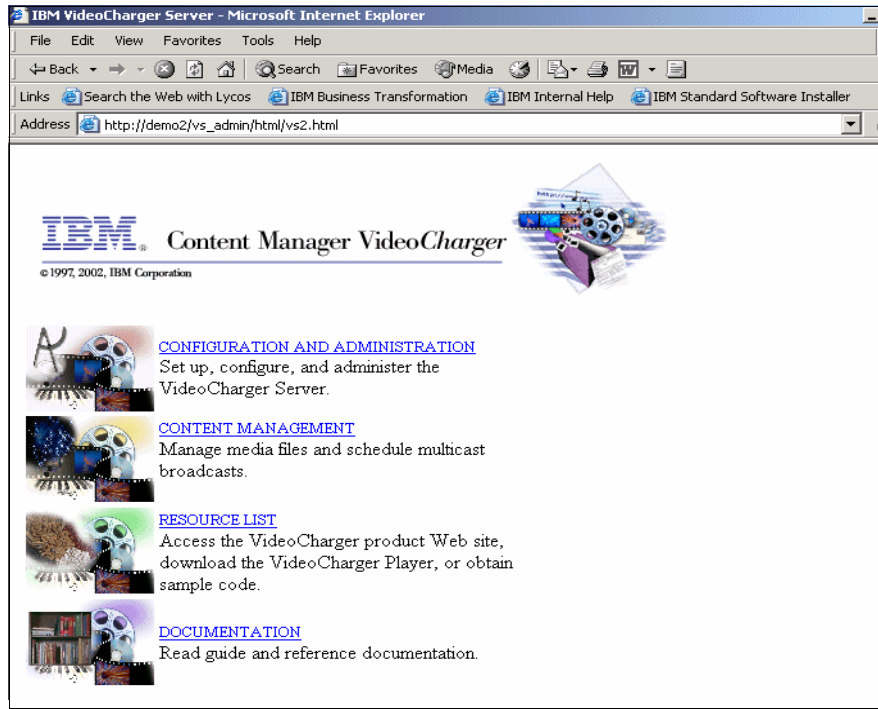


Figure 12-2 VideoCharger configuration and administration

After installation, you should also test to see if your VideoCharger is up and running:

1. On the VideoCharger page (see Figure 12-2), select **Content Management**.
2. Add the sample15.mpg video from your VideoCharger installation directory.
In our setup, it is:
C:\IBM\IBMVideoCharger Server\data\public\samples\content
3. On the Content Management advanced, Add Local Asset Files page, load it.
4. Run the video.
5. Open a Web browser and go to <http://localhost/lantv/>.
6. From the VideoCharger page, select **List Available Assets**, as shown in Figure 12-3 on page 259.
7. Select the video from the result list.
In our example, we name it IBMVideo.
8. The VideoCharger player should be automatically launched to play the selected video.



Figure 12-3 View and run the sample IBM video

If your selected video is not playing, troubleshoot your installation before you continue.

12.2 Integration of VideoCharger and Content Manager

VideoCharger has limitations on its indexing and searching capabilities. Because it integrates into Content Manager and is a component (a Resource Manager) of the overall Content Manager system (starting with V8), VideoCharger leverages the capabilities of Content Manager so that collectively, the overall system can handle the storage, archival, access control and delivery of videos. Applications such as the Content Manager Windows Client or a LWWCM integrated Web site can search for video titles, descriptions, or other metadata, and play the videos by launching the VideoCharger player from the Content Manager Windows Client or the Web browser. In addition, other video players can be configured to play the streamed videos so the client can open the appropriate video player and initiate a streaming session with the VideoCharger server.

One or more VideoCharger servers can be configured to one or more Content Manager Resource Managers. In this section, we describe in detail the steps to integrate the VideoCharger with our Content Manager system. The integration of the VideoCharger as a Content Manager Resource Manager allows you to stream media objects that are managed by Content Manager. In later sections, we will import some sample videos with the Content Manager Windows Client,

search for and play them to test the integration setup with the Content Manager Windows Client application as the prerequisite for the LWWCM Web publishing process integration.

12.2.1 Configuring a Resource Manager storage managed system

To integrate a VideoCharger with Content Manager, we must configure the Resource Manager storage managed system (SMS) and define a new collection to be used by the VideoCharger server. Figure 12-4 shows the SMS configuration for VideoCharger. An object storage volume is used. The VideoCharger volume represents a VideoCharger server asset group.

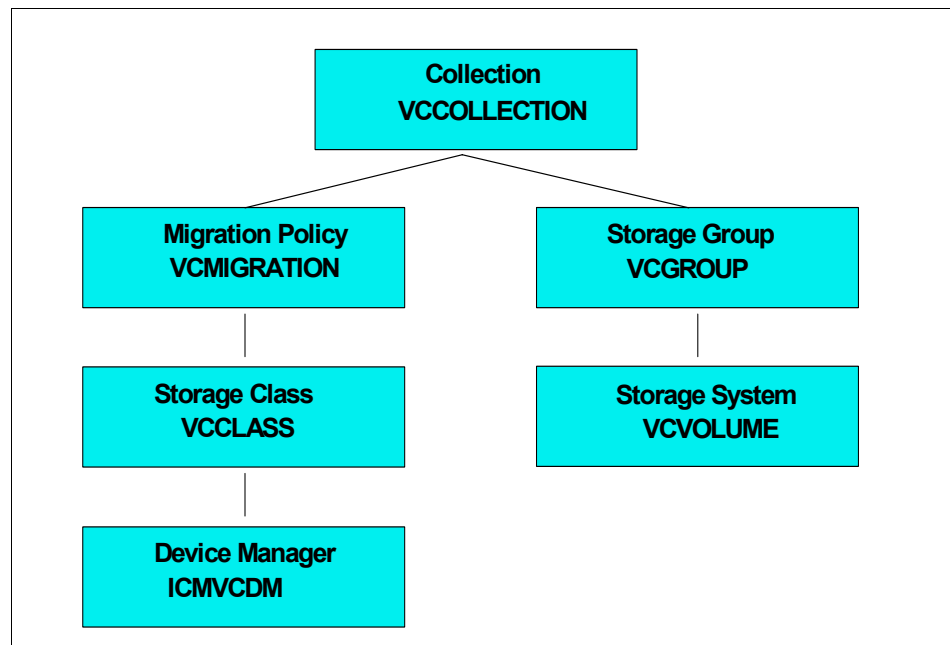


Figure 12-4 VideoCharger storage managed system configuration

We configure the SMS for VideoCharger with the Content Manager System Administration Client as follows:

1. Expand the Resource Manager from the defined Resource Manager systems. In our scenario, we choose to use our local Resource Manager system (RMDb).
2. Define a new media manager server by right-clicking **Server Definition** and selecting **New** from the context menu. In our scenario, we name the server definition VCSERVER (see Figure 12-5 on page 261).

3. Enable the VideoCharger device manager named ICMVCDM.
4. Define a new storage class, VCCLASS.
5. Define a new storage group, VCGROUP.
6. Expand the **Storage Systems** folder and define a new VideoCharger Volume VCVOLUME (see Figure 12-5).
7. Define a new migration policy name, VCMIGRATION. Select **forever** as the retention period.
8. Define a new collection, VCCOLLECTION (see Figure 12-5).

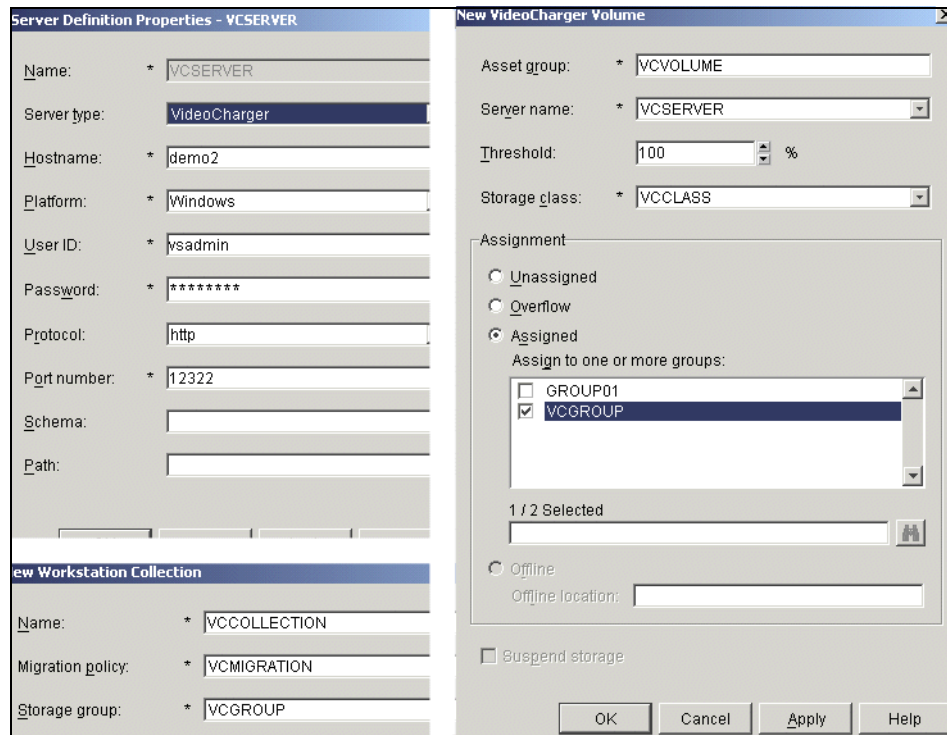


Figure 12-5 Set up VideoCharger SMS configuration

12.2.2 Creating a video insurance item type

Once we set up SMS configuration for VideoCharger, we need to create an insurance video item type which will use the collection VCCOLLECTION storage system:

1. Open Content Manager System Administration Client.

2. Open the data modeling folder. Right-click **Item Types** and select **New** from the context menu.
3. On the Definition tab, name your new document item type. In our scenario, we name it InsuranceVideo.
4. On the Attribute tab, add the attributes as shown in Figure 12-6. This metadata (attributes) associated with the VideoCharger assets is stored in the Library Server database ICMNLSDB.

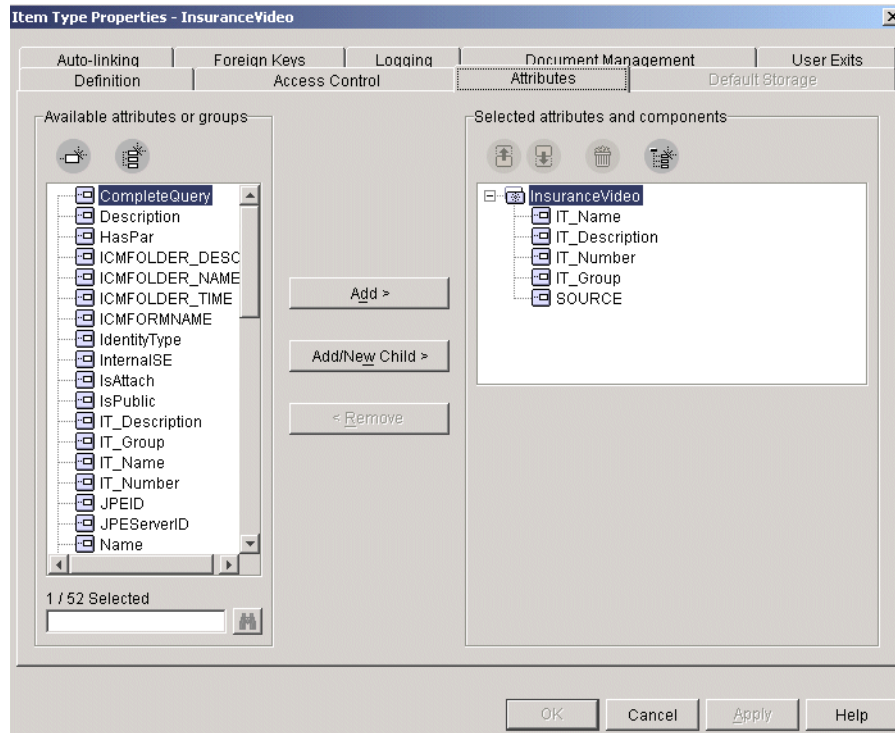


Figure 12-6 Selected attributes for the InsuranceVideo Item type

5. On the Document Management tab, add the ICMBASESTREAM part type, and choose the **PublicReadACL** access control list and the collection **VCCOLLECTION** as previously defined in the RMDB Resource Manager database. For the insurance video document item type, we use the predefined ICMBASESTREAM part type to handle the storing of video assets.

12.2.3 Importing, searching for and playing videos

Before integrating videos in your integrated LWWCM Web site, we need to test the Content Manager video system integration. This can be done by importing

some videos with the Content Manager Windows Client (see Figure 12-7). When we import a video, the insurance video attributes will be stored on the Library Server, ICMNLSDB, whereby the video file will be stored in the Resource Manager VideoCharger server. Depending on the size of the video you import, the used hardware and the network, the time it takes to load the video will vary. In our scenario, we set up on a Windows machine with 1.6 MB memory, and it takes 20 seconds to load the 8.5 MB sample15.mpg video file.

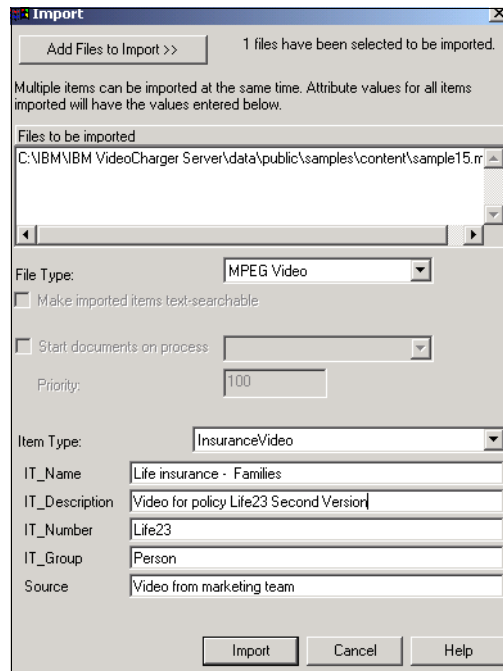


Figure 12-7 Importing some videos from Content Manager Windows Client

The media object loads onto the VCSERVER with an asset name of Lx.ABC.Vy where x represents the Library Server number, ABC represents a 26-character asset name, and y represents the version control number.

Note: If you have not installed the IBM License Use management software, you will receive an error when trying to import any video (see Figure 12-8).

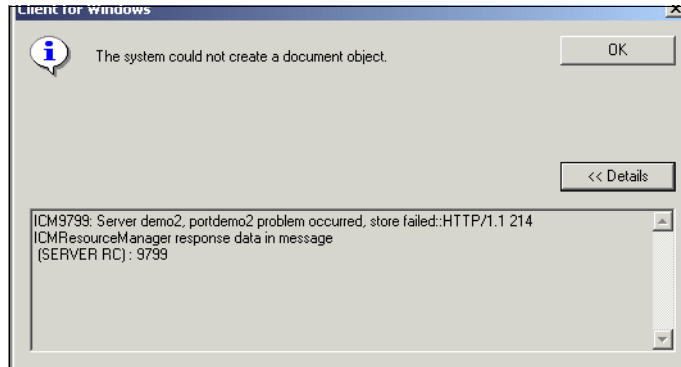


Figure 12-8 Error while importing a video if IBM LUM is not installed

Once the sample files are imported, we execute a basic search against the InsuranceVideo item type from the Content Manager Windows Client. Figure 12-9 shows the search results in our example.

InsuranceVideo					
	IT_Name	IT_Description	IT_Number	IT_Group	Source
	Life insurance - Families	Video for policy Life23 Second Version	Life23	Person	Video from marketin...
	Life insurance - Car driver	Video for policy Life43 First Version	Life3	Person	Video from departm...

Figure 12-9 Running a basic search against the InsuranceVideo item type

Before playing a video from the search result, Content Manager Windows Client must be configured to launch the appropriate VideoCharger player. For MPG files, as in our example, the configuration is done as follows:

1. Select **Options -> Preferences -> Helper Applications**.
2. Select the file type **MPEG Video** and enter the extension MPG.
3. Select the **Launch Application** check box.
4. In the Launch Application entry field, specify the location of the VideoCharger player in your setup. In our example, it is:
C:\program files\ibm\IBM VideoCharger Player\iscview.exe
5. Click **OK** to save the changes (see Figure 12-10 on page 265).

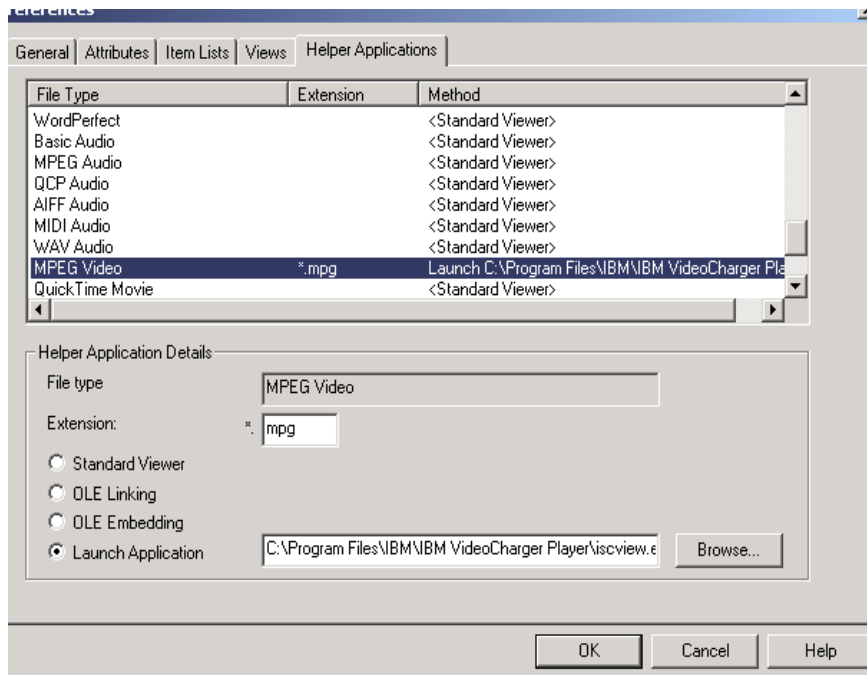


Figure 12-10 Launching the VideoCharger player application

Now that the helper application is configured, you can open or double-click an insurance item object from the basic search result. The VideoCharger player will be launched, and the video file will begin to be streamed from the VideoCharger server to the player. Figure 12-11 on page 266 shows the VideoCharger player playing the selected video.

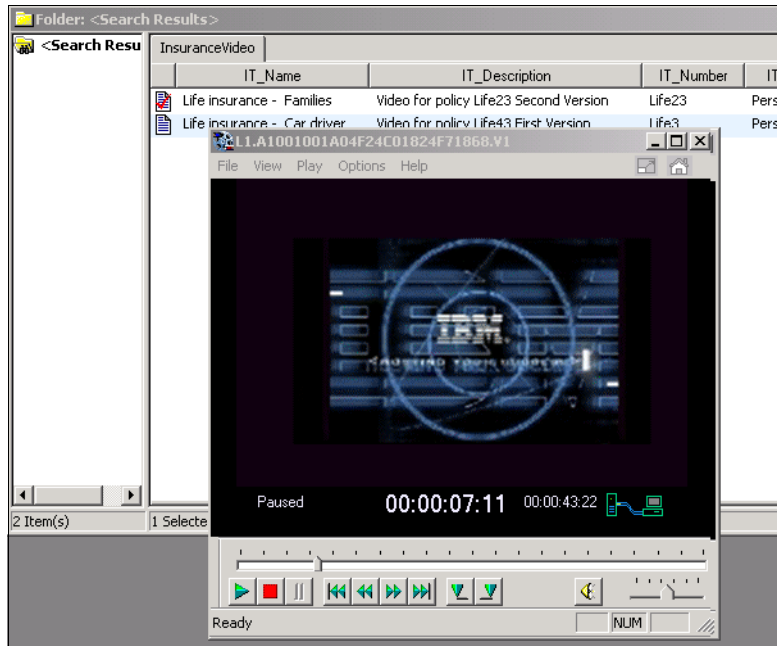


Figure 12-11 Open an InsuranceVideo item to play the video

The following steps describe what exactly happens when we request item objects from the InsuranceVideo item type:

1. The Content Manager Windows Client requests a media object from the Content Manager Library Server, ICMNLSDB.
2. The Library Server returns a security token and information where the video is stored on the icrm Resource Manager to the Content Manager Windows Client.
3. The client requests the sample15.mpg video media object from the Resource Manager.
4. The Resource Manager requests the video from our VideoCharger server, VCSERVER.
5. The VideoCharger server sends a temporary metadata file to the client, containing system instructions on streaming the media object.
6. The VideoCharger player is launched and the streaming session is initiated.
7. The video is streamed from the server to the player.

12.3 Integrating Content Manager videos to a Web site

In this section, we show you how to integrate Content Manager videos to your integrated LWWCM Web site.

12.3.1 Why integrate Content Manager videos to a Web site?

Sometimes it can be necessary or interesting to show video in your Web site, particularly demonstration videos. For example, referring to our case study discussed in Chapter 4, “Case study” on page 41, it could be necessary for the insurance company to include some video showing how important it is to have insurance protecting you if/ when you have an accident.

There are many other examples of reasons to integrate video in your Web site:

- ▶ Broadcast a CEO’s speech about the future directions and strategies of the company, through the intranet to the company’s employees who are located all around the world.
- ▶ For educational purposes, stream video classes over a secure Internet.
- ▶ Launch an advertising campaign with new videos about the product on the Internet.
- ▶ Replay sports games or TV programs.

How do you integrate Content Manager videos to your Web site? We show you the steps by demonstrating using our case study.

12.3.2 Prerequisites

VideoCharger must be properly installed as explained in 12.2, “Integration of VideoCharger and Content Manager” on page 259. We also need some additional configuration from Information Integrator for Content in order to achieve the integration of Content Manager video with the LWWCM Web site.

Note: Remember to restart the LWWCM server after installing VideoCharger.

Since videos are stored directly in the Content Manager repository, we have to create a new federated entity and a new search template, whereby LWWCM can find and retrieve the videos. Use the same steps described in “What we need from Information Integrator for Content” on page 125 to map the VideoCharger item type attributes created earlier (see Figure 12-6 on page 262) with the federated attributes.

The new search template will be used to search and retrieve the videos from Content Manager to LWWCM.

12.3.3 Creating video components in LWWCM

The video files are stored within Content Manager and are associated with a specific item type (in our case study, the item type is *InsuranceVideo*). We can retrieve them in the same way we do any other types of content, as explained in Chapter 7, “Publishing Content Manager content” on page 115. To achieve this, we have three options: using federated content components, using JSP components or using connect tags.

For demonstration purposes, we show you how to integrate video files using the first option, *federated content components*.

We need to create a new federated content component (VC_Component) in LWWCM to retrieve the stored video files from Content Manager (VideoCharger). Follow the steps detailed in 7.3, “Federated content component” on page 123 to create the new federated content component for video purposes.

Figure 12-12 shows the video component, VC_Component, which is really a federated content component, which we created in LWWCM.

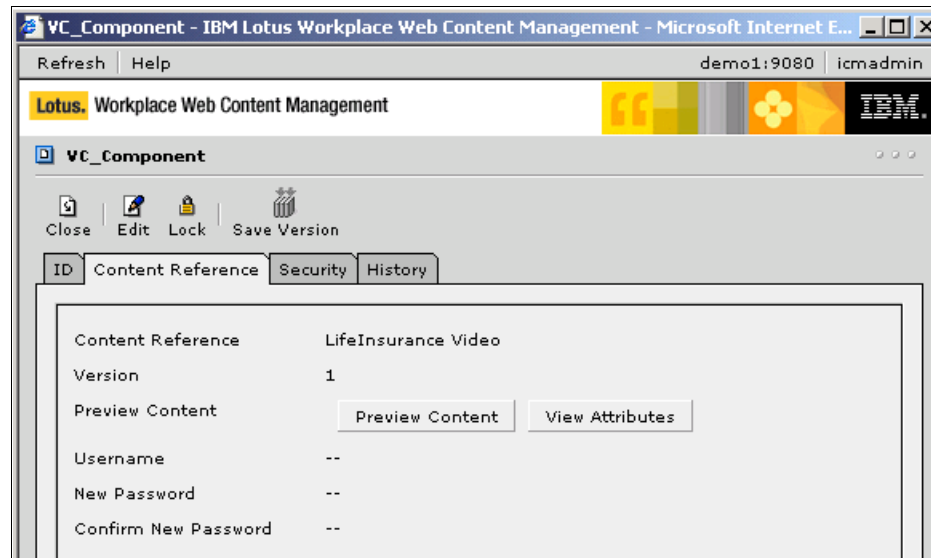


Figure 12-12 Federated content component for video search

12.3.4 Incorporating video within a Web page

To integrate the videos within a LWWCM Web page, we must go to the content page where we want to display videos and add the newly created federated component in the Content tab. In our case study, we called this federated content component within the content object VC_Component. We need to include this name in the proper tag parameters to retrieve the video file.

There are two main ways to integrate videos within a Web page:

- ▶ **Embedded videos:** Show the video embedded in the content of the Web page.
- ▶ **Video links:** Provide a link to the video URL, which the user can click to display the video.

Important: The *Scope* parameter within the tag *must* be contentURL.

Embedded videos

To show how a video is embedded within a Web page, we provide sample code in Example 12-1. This code must be included in the *page design object* which is associated with the target Web page. Remember that the association between Web pages and page design objects is determined in the proper site area entry, in the site framework.

Example 12-1 How to embed a video within a Web page

```
<table height=100% width=100%> <tr> <td align="center">
<FederatedCmpnt context="current" end="'&gt;" key="VC_Component"
Scope="contentUrl" start="&lt;EMBED width=100% height=300 SRC=''"
type="content"/> </EMBED></td></tr></table>
```

Video links

Sometimes, it might be necessary to supply a list of video links or perhaps just a single link to a specific video instead of displaying the video directly within a Web page. In this case, we can choose to display a link with the video file URL, and users can click the link to display the video. Again, we have to modify the proper page design object to determine how to display this information in the Web browser. We provide sample code in Example 12-2.

Example 12-2 How to show only the video link

```
<b>Video file:</b> <FederatedCmpnt context="current" end="'&gt;"
key="VC_Component" Scope="contentUrl" start="&lt;a target=_blank href=''"
type="content"/>Click here for the video file</a><br>
```

The easiest way to show a list of video links is to use the *connect tags* or *JSP components*, since the federated content component only references *one* video file. For further information on how to show lists of external resources, refer to 7.5, “JSP component” on page 151.

Note: The application which plays the video within your Web page is the one associated with the MIME type in the browser configuration. By default, in Windows operating systems, this application is the Windows Media player.

Embedding video using the federated content component is a matter of creating the proper federated content component and referencing it in the page design using the proper tag and attributes. There is actually no difference with embedding other content using the federated content component, except the way the retrieved URL is used in the HTML -> EMBED tags. If the video does not appear on the screen, it is probably a matter of having the proper video player plugin installed or a wrong MIME type reference.



Data migration through syndication

This chapter discusses the different alternatives for LWWCM data migration. Since the way to achieve this is through the syndication tool that LWWCM provides, we also describe the syndication features.

We cover the following topics in this chapter:

- ▶ Syndication overview
- ▶ Migrating content repositories
- ▶ Upgrading LWWCM V1.1 content to LWWCM V2.0

13.1 Syndication overview

Syndication is a way of replicating (copying) data from one server to another, thus making the content available across many Web sites over many servers. LWWCM uses syndication to push or pull content from different environments to replicate data from one server to another. The back-end repository for the servers do not have to be same. This means that it is possible to replicate a Web site's data that uses file system as the back-end repository to a Web site that uses Content Manager as the back-end repository.

Normally, a company's Web sites use several servers to set up data. You have development, staging, and production servers. These servers may need to share data among themselves. Syndication is the tool to make this possible.

Syndication provides two different objects, the syndicator and the subscriber.

Note: The *syndicator object* is the *source* site where the data is coming from. The *subscriber object* is on the *destination* site where the data is going to.

13.1.1 Setting up the syndication tool

The syndicator and the subscriber are related. Their relationships may be in both ways, one-way and two-way transportation. *One-way transportation* is the most common approach. To achieve this, a syndicator object must be created in the source server and a subscriber object in the destination server. To make the *two-way transportation* possible, a syndicator and subscriber's relationship must be created on both servers.

To enable the syndication on the servers, you must first set some parameters in the `aptrixje.properties` file. Refer to the LWWCM's online Information Center to get the detailed information (**Installation Guide -> Syndication -> Enabling Syndication**).

Example 13-1 shows a sample of configuration for our environment.

Example 13-1 Sample configuration in `aptrixje.properties`

```
# deployment settings
deployment.itemDispatcherUrl=http://90.30.40.219:9080/itjumpstart/connect/?MOD=
ItemDispatcher
deployment.syndicatorUrl=http://90.30.40.219:9080/itjumpstart/connect/?MOD=Synd
deployment.subscriberUrl=http://90.30.40.219:9080/itjumpstart/connect/?MOD=Subs
```

Creating the syndicator

To create a syndicator:

1. Open the LWWCM user interface.
2. Select **Syndication & Deployment** -> **Syndicator** from the left navigator.
3. Click **New**.
4. Enter appropriate values for the ID and Syndicator section.
5. Click **Save**.

Creating the subscriber

To create a subscriber:

1. Open the LWWCM user interface.
2. Select **Syndication & Deployment** -> **Subscriber** from the left navigator.
3. Click **New**.
4. Enter appropriate values for the ID and Subscriber section.
5. Click **Save**.

Once we have created the syndicator and the subscriber, we must relate both objects to perform syndication.

Important: The syndicator object must always be saved *before* the subscriber object.

To establish the relationship between the syndicator and the subscriber, you need to set the information on the second tab of each object. In the syndicator object, in the Syndicator tab, the information about the subscriber is required. On the other hand, in the subscriber object, in the Subscriber tab, the information about the syndicator is required.

The following are the syndicator and subscriber mandatory fields required to set up the syndication:

- ▶ In the syndicator form:
 - **Syndicator ID** and **Syndicator URL**: These fields are automatically computed by the system. You need to copy and paste them into the relevant fields in the subscriber object.
 - **Subscriber Name**: Name to describe the subscriber. Copy it from the subscriber form and paste it here.
 - **Subscriber ID** and **Subscriber URL**: Copy the subscriber ID and the subscriber URL from the subscriber form and paste them here.
 - **Item Gatherer**: This item has two options:
 - All Items: Gathers a collection of all deployable items.

- All Live Items: Gathers a collection of all deployable items that either do not have a workflow or do have a workflow and are published.
- In the the subscriber form:
- **Subscriber ID** and **Subscriber URL**: These fields are automatically computed by the system. You need to copy and paste them into the relevant fields in the syndicator object.
 - **Syndicator Name**: Name to describe the syndicator.
 - **Syndicator ID** and **Syndicator URL**: Copy the syndicator ID and the syndicator URL from the syndicator form and paste them here.
 - **Current State**: Displays the current state's reference. If syndication is working, both the syndicator and subscriber should have the same current state.

In Figure 13-1, we put both the syndicator form and subscriber form together and show how this information is filled in to set the syndication process.

The image displays two web forms side-by-side. The left form, titled 'ServerDemo1', is the 'Syndicator' form. It contains fields for 'Syndicator Id' (e229334246732a7), 'Syndicator Url' (http://demo1:9080/ilwcm/connect/?MOD=Synd), 'Subscriber Name' (ServerSusDemo2), 'Subscriber Id' (49a2935424777ce8), 'Subscriber Url' (http://demo1:9080/ilwcm/connect/?MOD=Synd), 'Item Gatherer...' (All Items), 'Current State' (876), 'Item Formatter' (Java Serialisation Formatter), 'Enabled' (checked), and 'Current Status' (Idle). The right form, titled 'Subs ServerDemo2', is the 'Subscriber' form. It contains fields for 'Subscriber Id' (49a2935424777ce8), 'Subscriber Url' (http://demo1:9080/ilwcm/connect/?MOD=Subs), 'Syndicator Name' (ServerDemo1), 'Syndicator Id' (e229334246732a7), 'Syndicator Url' (http://demo1:9080/ilwcm/connect/?MOD=Synd), 'Current State' (ICE-INITIAL), 'Enabled' (checked), and 'Current Status' (Idle). Red arrows point from the Syndicator form to the Subscriber form, and blue arrows point from the Subscriber form back to the Syndicator form, illustrating the cross-referencing of information.

Figure 13-1 Crossed information between Syndicator and Subscriber

Once the information given above is set up, we can continue with the syndication process.

The last step is to activate the syndication. This is possible by entering the subscriber object and clicking the **Subscriber Rebuild** button:

1. Open the LWWCM user interface.
2. Select **Syndication & Deployment -> Subscriber**.
3. Double-click the subscriber object.

4. Click **Subscriber Rebuild**.

After a few minutes, the data will be gathered to the subscriber LWWCM application.

For more information about syndication, refer to the LWWCM's online Information Center.

13.1.2 Using unique Content Manager repository

Now that we have described how syndication works, we will study the different types of data repositories for syndication.

We have the possibility of storing data from two different LWWCM applications into the same Content Manager repository. To accomplish this configuration, we must ensure that the `aptrixjpe.properties` files are properly set up in both LWWCM servers pointing to the same Content Manager as the data repository as shown in Example 13-2.

Example 13-2 aptrixjpe.properties configuration sample

```
manager.persistence=CM
resource.persistence=CM
cm.server=[CM_SERVER_NAME]
cm.username=[CM_USERNAME]
cm.password=[CM_PASSWORD]
cm.jpserverIdentifier=[CM_ILWWCM_SERVER_ID]
```

In this case, because we use more than one LWWCM application server with a single Content Manager server, each LWWCM application server must have a unique server ID (see text in bold in the example).

Once the `aptrixjpe.properties` file is set up, you need to do the following to enable syndication:

1. Create a new syndicator on server 1.
2. Create a new subscriber on server 2.
3. Syndicate. Click **Subscriber Rebuild** in the subscriber object in server 2.

After the syndication, the data in the syndicator server and in the subscriber server will be synchronized. Their data will be stored in Content Manager with different server ID identifying the source of the data. See Figure 13-2 on page 276 as an example.

Folder: <Search Results>				
AJPEData				
	Name	JPE Server ID	Type	JPE ID
	Education and Training	demo3	contents	e50049033c30f0d
	Education and Training	1	contents	e50049033c30f0d
	Education and Training	demo2	contents	e50049033c30f0d
	Education Policy	demo3	contents	e5000c024dbe614
	Education Policy	1	contents	e5000c024dbe614
	Education Policy	demo2	contents	e5000c024dbe614
	Employee Exit Information	demo3	contents	e5000900beb001e
	Employee Exit Information	1	contents	e5000900beb001e
	Employee Exit Information	demo2	contents	e5000900beb001e
	Employee Handbook	demo3	contents	e50006024d260fb
	Employee Handbook	1	contents	e50006024d260fb
	Employee Handbook	demo2	contents	e50006024d260fb

Figure 13-2 Content Manager AJPEData table

The example shows LWWCM data coming from three different LWWCM servers, 1, demo2, and demo3.

This does not mean that they always have to depend on the syndicators. After all sites are replicated, you can always uncheck the **Enabled** field to disable any future replication transactions. See Figure 13-3 for details.

SubsServerDemo2

Close

Save

Save As...

ID

Subscriber

Security

History

Subscriber Id

49a2935424777ce8

Subscriber Url

http://demo1:9080/ilwwcm/connect/?MOD=Subs

Syndicator Name

ServerDemo1

Syndicator Id

e229334246732a7

Syndicator Url

http://demo1:9080/ilwwcm/connect/?MOD=Synd

Current State

ICE-INITIAL

Enabled

☐

Current Status

Disabled

Figure 13-3 Subscriber enable option

In Figure 13-4 on page 277, we show the architecture of using one Content Manager as repository for two LWWCM servers that are syndicator and subscriber.

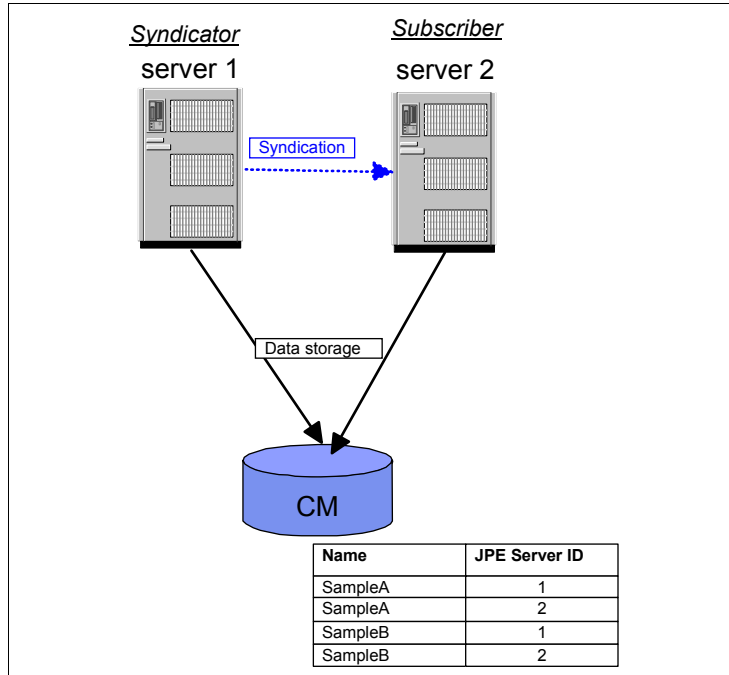


Figure 13-4 Syndication using the one Content Manager repository

Note that, in the beginning, the data in server1 and server2 is stored in Content Manager as separate content. It is the same in the beginning; this implies duplicate data in Content Manager. With this configuration, however, you have the flexibility of disabling the syndication later and making each site unique.

13.1.3 Using different repositories

Syndication can be used not only to transport data between servers, but also between different types of repositories.

In this case, for example, we can have syndication between a server using the file system as the data repository with another server using Content Manager as repository. The goal for this type syndication is to migrate all the site information from the file system to Content Manager.

In this example, we assume:

- ▶ Both applications are running
- ▶ *Server 1* is the *host* of the data using file system as data repository
- ▶ *Server 2* is the *receiver* using Content Manager as data repository.

With this scenario, we do not have to change anything in the `aptrixjpe.properties` files. The required steps for this syndication of different repository is exactly same as the steps described earlier:

1. Create a new syndicator object on server 1.
2. Create a new subscriber object on server 2.
3. Syndicate by clicking **Subscriber Rebuild** from the subscriber object.

After syndication, the data will appear identical in both servers but in different repositories. The original data from server 1 remains untouched in the file system. A copy of this data in the Content Manager repository accessible by server 2.

Note: Rebuild sends all of the items while update only send the items that have changed since the last syndication over to the subscriber.

Figure 13-5 describes this scenario.

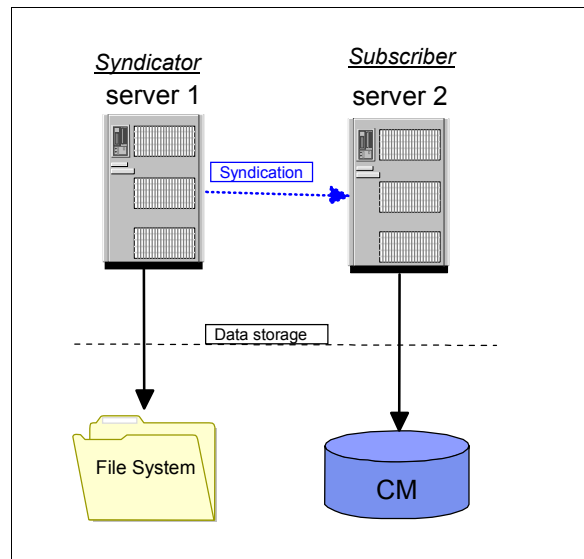


Figure 13-5 Syndication using two different repositories

13.2 Migrating content repositories

After you change the persistency setting in the `aptrixjpe.properties` file, the next time you start up the LWWCM server, the LWWCM server will try to find its data in the new repository. Changing the persistency settings only *does not* move the

data to the new repository. In order to move the data from one repository to another, you have to create a temporary LWWCM server and syndicate the data to this temporary server. After this, you configure the original server to point to the new repository and the temporary server can be deleted. Figure 13-6 illustrates this data migration approach using the file system to Content Manager as an example.

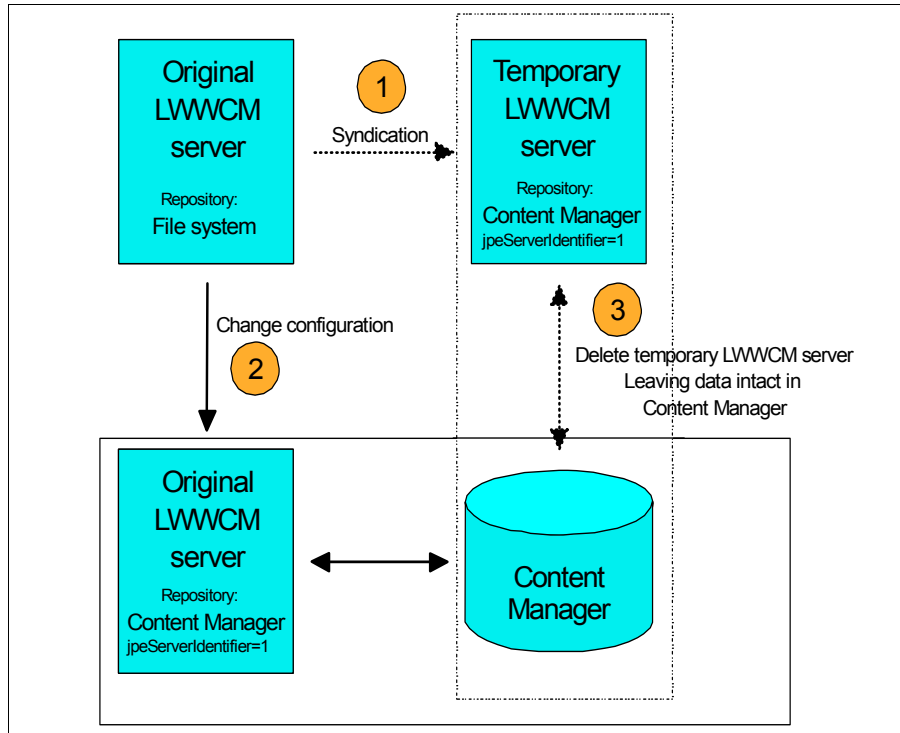


Figure 13-6 Migration approach

In this section, we discuss the necessary steps to accomplish a successful data migration from an existing LWWCM file system repository to a repository of type Content Manager. This is all done on one box using one application server (server1); hence the original and temporary LWWCM servers are accessible on the same port (here:9080) but in a different context:

- ▶ Original server: <http://demo99/ilwcm/connect/intranet>
- ▶ Temporary server: http://demo99/ilwcm_temp/connect/intranet

Of course, you can use different machines or different application servers on the same machine to achieve the migration task. Using these steps enables you to seamlessly migrate the content repositories without having users changing their normal ways of accessing the LWWCM Web site.

The steps are summarized as follows:

1. Clean up your original system.
2. Back up your existing data.
3. Create a temporary LWWCM server using CM persistence (Step 1 in Figure 13-6 on page 279).
4. Syndicate the temporary LWWCM server (Step 1 in Figure 13-6 on page 279). This is done so that the temporary LWWCM server now has all the data (stored in Content Manager) from the original LWWCM server.
5. Configure the original server to use Content Manager repository (Step 2 in Figure 13-6 on page 279). This sets the original LWWCM server to use Content Manager as data repository instead of the file system. Note that, at this point, Content Manager contains all the data that the original server has.
6. Delete the temporary LWWCM enterprise application (Step 3 in Figure 13-6 on page 279).

Clean up your original system

Make sure that your original system is consistent to prevent syndication problems. Run the following checker tasks from your Web browser:

```
http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpeuserchecker&create=true  
http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpereferencechecker&fix=true  
http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajperesourcechecker&fix=true  
http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpesitechecker&fix=true
```

Back up your existing data

If you are using the file system as your current repository type, back up the following folders:

```
<LWWCM_HOME>\ilwwcm\data  
<LWWCM_HOME>\connect\users
```

<LWWCM_HOME> is the directory where you installed LWWCM.

Create a temporary LWWCM server using CM persistence

You can create a dedicated application server or use the same application server for the temporary LWWCM enterprise application. In our scenario, we use the same application server (server1) for both the original and the temporary LWWCM application. Our temporary application is called `ilwwcm_1_war`.

Table 13-1 on page 281 shows the assumptions made for this scenario.

Table 13-1 Variable and path assumptions

Variable	Value
<LWWCM_HOME>	c:\lwwcm_temp\app
<WEB_APP_HOME>	c:\ibm\WebSphere\AppServer\installedApps\<NODE>\ilwwcm_temp_war.ear\ilwwcm.war
<WAS_HOME>	c:\ibm\WebSphere\AppServer

The following list briefly describes the steps to set up a new LWWCM server using Content Manager as the repository (for detailed information, refer to Chapter 6, “Using CM as a repository for LWWCM” on page 91):

1. Increase the heap size of server1:
 - a. Open the WebSphere Administrative Console.
 - b. Goto **Application Servers -> server1 -> Process Definition -> Java Virtual Machine**.
 - c. Specify Initial Heap Size: 256
 - d. Specify Maximum Heap Size: 512
2. Extend the classpath of server1 with the Information Integrator for Content JAR files. Install Information Integrator for Content (EIP) Development Workstation on the machine where LWWCM is running:
 - a. Open the WebSphere Application Server Administrative Console.
 - b. Goto **Application Servers -> server1 -> Process Definition -> Java Virtual Machine**.

As a minimum, install the local connectors Content Manager V8 (ICM connector) and federated subcomponent (FED connector) on the machine where LWWCM is running. This installation installs all JAR files we specify in step 3 below. It is also useful to install the Connector Toolkits and Samples Content Manager V8 and Federated subcomponents, although they are not required for the following steps.

3. Add the following directory, JAR, and ZIP files with their path to the Classpath (the path may differ from that of your installation):

```
C:/IBM/CMgmt
C:/CMBROOT/LIB/cmbview81.jar
C:/CMBROOT/LIB/cmb81.jar
C:/CMBROOT/LIB/cmbcm81.jar
C:/CMBROOT/LIB/cmbjdbc81.jar
C:/CMBROOT/LIB/cmbfed81.jar
C:/CMBROOT/LIB/cmbdb281.jar
C:/CMBROOT/LIB/cmbdj81.jar
```

```

C:/CMBROOT/LIB/cmbicm81.jar
C:/CMBROOT/LIB/cmbutil81.jar
C:/CMBROOT/LIB/cmbutiljdbc81.jar
C:/CMBROOT/LIB/cmbutilicm81.jar
C:/CMBROOT/LIB/cmbutilfed81.jar
C:/CMBROOT/LIB/xerces.jar
C:/CMBROOT/LIB/essrv.jar
C:/CMBROOT/LIB/esclisrv.jar
C:/CMBROOT/LIB/cmblog4j81.jar
C:/CMBROOT/LIB/log4j.jar
C:/CMBROOT/LIB/cmbsdk81.jar
C:/CMBROOT/LIB/cmbwas81.jar
C:/IBM/SQLLIB/java/db2java.zip

```

4. Copy the LWWCM home (app) directory to <LWWCM_HOME>. This can be any directory you choose for LWWCM software.
5. Use the Application Assembly Tool to adjust the initialization parameters in the war file, c:\lwwcm_temp\app\config\websphere\ilwwcm.war. See Figure 13-7. Adjust the parameter values for ConfigFile parameter.

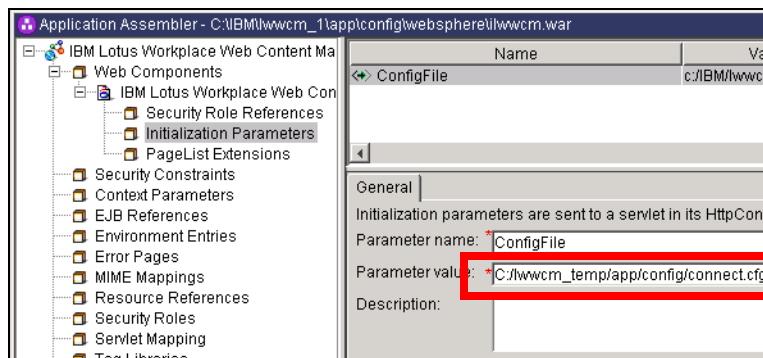


Figure 13-7 Application assembly for temporary LWWCM server

6. Install the temporary enterprise application:
 - a. Open the WebSphere Application Server Administrative Console.
 - b. Go to **Applications -> Install New Application**.
 - c. Browse to where the ilwwcm.war file is located. In our scenario, it is:
c:\lwwcm_temp\app\config\websphere\ilwwcm.war.
 - d. Specify Context Root: ilwwcm_temp.
 - e. Click **Next** until *Step 1: Provide options to perform the installation*.
 - f. Specify the Application Name: ilwwcm_temp_war.
 - g. Click **Next** until *Step 3: Map modules to application servers*.

- h. Map the new application to server1.
 - i. Click **Next**, **Finish**, **Save to Master Configuration**, and then **Save**.
7. Copy the LWWCM Web application to <WEB_APP_HOME>.
 8. Configure the persistence settings of the temporary server in aptrixjpe.properties (which can be found in <LWWCM_HOME>\config) to match the target repository type:


```
manager.persistence=CM
resource.persistence=CM
cm.server=ICMNLSDDB
cm.username=icmadmin
cm.password=<password>
cm.jpeServerIdentifier=demo99
cm.versionControl=false
```
 9. Configure any other Content Manager related settings in aptrixjpe.properties of the temporary server. A list of available options and their impact on the system can be found in LWWCM's online Information Center (**Installation Guide -> Server Configuration -> Web Content Management Data Storage -> Storing Data in IBM Content Manager**).
 10. Restart server1.


```
<WAS_HOME>\bin\stopserver server1
<WAS_HOME>\bin\startserver server1
```

<WAS_HOME> is where you installed WebSphere Application Server.
 11. Create the basic LWWCM user interface entries for the temporary server:
 - a. Enter the following URL in your browser:


```
http://<server>/ilwwcm_temp/connect/?MOD=AJPEGUI
```
 - b. Specify User name: Administrator.
 - c. Specify Password: password.

Syndicate the temporary LWWCM server

Syndicate the content from the original LWWCM server to the temporary LWWCM server. This moves the content from the original LWWCM server into the repository of the temporary server, which is configured to be Content Manager.

To perform the syndication, create a new subscriber on the temporary LWWCM server and a new syndicator on the original LWWCM server and start syndication by following the steps described in 13.1.1, "Setting up the syndication tool" on page 272. Figure 13-8 on page 284 shows the creation of the subscriber and the syndicator.

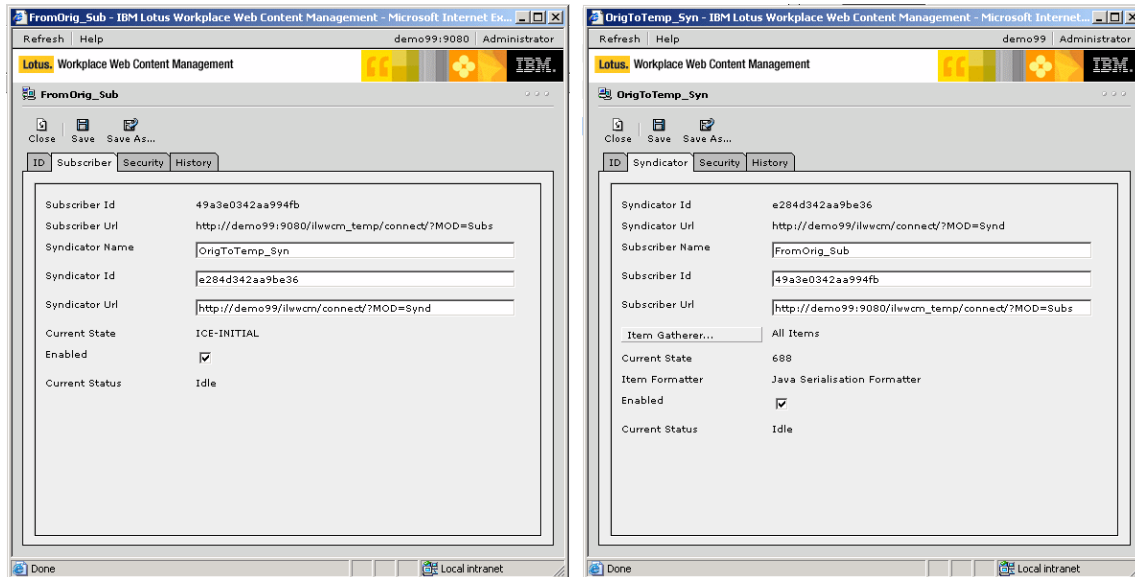


Figure 13-8 Syndicator and subscriber for migration

After successful syndication, you should test if the data has been moved properly by entering the following URL:

`http://<server>/ilwcm_temp/connect/Intranet`

This opens the ITJumpstart Intranet Page on your temporary LWWCM server. Additionally, you can verify that the data and the resources are created in your Content Manager. To check this, do the following:

1. Log in to your Content Manager System Administration Client.
2. Search the item type AJPEData. Leave all search fields empty. The result should look similar to Figure 13-9. Note the number of returned items and the JPE Server ID column.

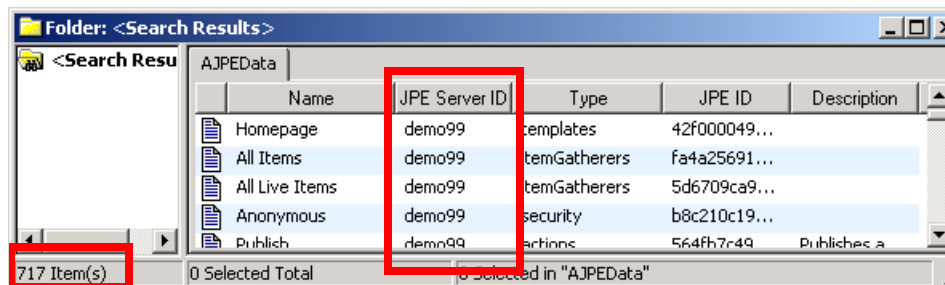


Figure 13-9 AJPEData search result in Content Manager after syndication

If everything is working properly, you can delete the syndicator and subscriber entries from the original and temporary LWWCM server. This step is optional.

Configure the original server to use Content Manager repository

The following steps are required to configure the persistence settings or the original server to use Content Manager as the repository:

1. Change the following entries in the `aptrixjpe.properties` file which can be found in the `<LWWCM_HOME>\config` directory. Make sure that all Content Manager related entries match the configuration of the temporary server:

```
manager.persistence=CM
resource.persistence=CM
cm.server=ICMNLSDDB
cm.username=icmadmin
cm.password=<password>
cm.jpeServerIdentifier=demo99
cm.versionControl=false
```

2. Configure any other Content Manager related settings in `aptrixjpe.properties` of the original server. Make sure that all entries match the configuration of the temporary server.
3. Configure the temporary server to use the file system as its repository to prevent the temporary server from using Content Manager during the next `server1` startup. Change the following two lines in the `aptrixjpe.properties` file of the temporary server:

```
manager.persistence=FileSystem
resource.persistence=FileSystem
```

Note that the system path contains system related informations and the data path contains all LWWCM data files.

4. Shut down the original and temporary LWWCM server. Since both applications are installed on `server1`, you have to stop `server1` by executing the following command in the WebSphere Application Server bin directory:

```
<WAS_HOME>\bin\stopserver server1
```

5. Delete the following folders and their content from the original server:

```
<LWWCM_HOME>\data
<LWWCM_HOME>\system
```

6. Copy the following folder and its content from the temporary to the original server:

```
<LWWCM_HOME>\system
```

7. Restart the server which holds the original LWWCM application to activate the new configuration.

```
<WAS_HOME>\bin\startserver server1
```

Delete the temporary LWWCM enterprise application

To delete the temporary LWWCM enterprise application, do the following:

1. Open the WebSphere Application Server Administrative Console.
2. Go to **Applications -> Enterprise Applications**.
3. Check the application you want to delete. In our case, it is ilwwcm_1_war.
4. Click **Stop** (see Figure 13-10).
5. Click **Uninstall** -> **Save** -> **Save**.

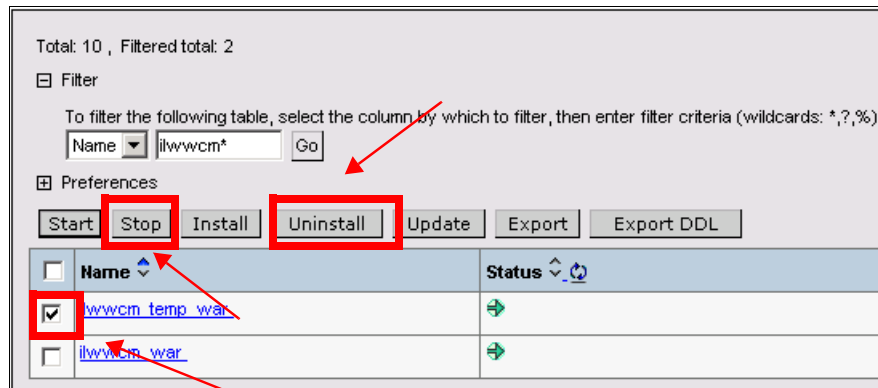


Figure 13-10 Deleting the temporary LWWCM application

13.2.1 Troubleshooting

If you cannot access the original system after migration, do the following to solve the problem:

1. Stop server 1:
`<WAS_HOME>\bin\stopserver server1`
2. Delete the users folder:
`<LWWCM_HOME>\connect\users`
3. Start server 1:
`<WAS_HOME>\bin\startserver server1`
4. Run the LWWCM checker utilities from your browser:
`http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpeuserchecker&create=true`
`http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpereferencechecker&fix=true`
`http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajperesourcechecker&fix=true`
`http://[HOST]:[PORT]/[CONTEXT]/connect?mod=ajpesitechecker&fix=true`

5. Stop server 1:
`<WAS_HOME>\bin\stopserver server1`
6. Delete the following folder:
`<LWWCM_HOME>\system`
7. Start server 1:
`<WAS_HOME>\bin\startserver server1`

13.3 Upgrading LWWCM V1.1 content to LWWCM V2.0

In this section, we address the scenarios in which you need to upgrade an existing LWWCM V1.1 server and its content to a LWWCM V2.0 server using the following two paths:

- ▶ File system to file system upgrade
- ▶ File system to Content upgrade/migration

13.3.1 File system to file system upgrade

You need to upgrade the LWWCM Web application with V2.0 and reuse the data and user folders from V1.1. There is no need for data migration. After that, you can generate a new V2.0 LWWCM user interface.

Upgrade the LWWCM Web application

To upgrade the LWWCM Web application:

1. Open the WebSphere Application Server Administrative Console.
2. Goto **Applications -> Enterprise Applications**.
3. Check the LWWCM Web application. In our scenario, it is `ilwwcm.war`.
4. Click **Stop**. Wait until the application is stopped and click **Update** (see Figure 13-11 on page 288).

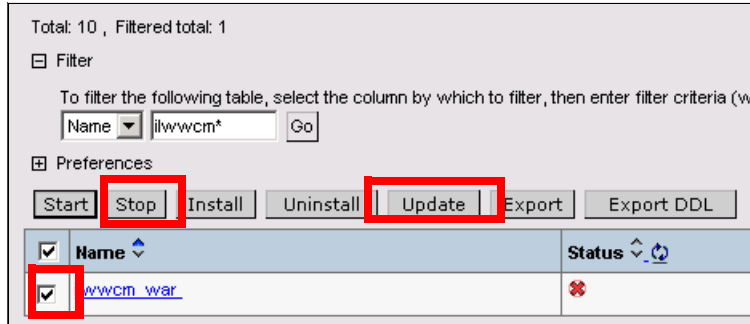


Figure 13-11 Updating the LWWCM Web application

5. From your installation CD, copy the web-app folder to <WEB_APP_HOME>:
c:\ibm\WebSphere\AppServer\installedApps\demo99\ilwwcm_war.ear\ilwwcm.war
6. Go back to the WebSphere Application Server Administrative Console; go back to **Applications -> Enterprise Applications**.
7. Check the LWWCM Web application (ilwwcm.war).
8. Click **Start**. Wait until the application is started and click **Update**.
9. Run the LWWCM resource checker utility. This creates the resource folder in the <WEB_APP_HOME> directory:

<http://demo99/ilwwcm/connect/?MOD=AJPEResourceChecker&FIX=TRUE>

Generate the new LWWCM V2 user interface

Execute the following command line in the browser:

[http://\[HOST\]:9080/ilwwcm/connect/?MOD=AJPEGUI](http://[HOST]:9080/ilwwcm/connect/?MOD=AJPEGUI)

13.3.2 File system to Content Manager migration

Upgrading a V1.1 file system repository to a V2.0 Content Manager repository is a combination of 13.2, “Migrating content repositories” on page 278 and 13.3.1, “File system to file system upgrade” on page 287:

1. You have to upgrade the LWWCM Web application as shown in 13.3.1, “File system to file system upgrade” on page 287.
2. Create a temporary LWWCM V2.0 server configured to use Content Manager as the repository. Then syndicate the content to the temporary server, switch the configuration of your original LWWCM server to use Content Manager as the repository. The detailed steps are discussed in 13.2, “Migrating content repositories” on page 278.

13.3.3 Enabling LWWCM version control

You do not have LWWCM's version control on the migrated system by default because this is a new feature of LWWCM V2.0 and your content was created using LWWCM V1.1. You can, however, enable the version control for the new contents as follows:

1. Open the `aptrixjpe.properties` file which can be found in the `<LWWCM_HOME>\config` directory.
2. At the `# control properties` section, add the value to each object that requires version control:

```
com.aptrix.versioncontrol.VersioningControl
```

Example 13-3 shows a sample of the `aptrixjpe.properties` file that enables version control.

Example 13-3 aptrixjpe.properties file: enable version control

```
# control properties
control.Content=com.aptrix.versioncontrol.VersioningControl,com.aptrix.pluto.workflow.WorkflowControl,com.aptrix.pluto.taxonomy.ProfileControl
control.Style=com.aptrix.versioncontrol.VersioningControl
control.Template=
control.Taxonomy=
control.Category=
control.Site=
control.SiteArea=
control.Workflow=
control.WorkflowStage=
control.WorkflowAction=
control.Cmpnt=com.aptrix.versioncontrol.VersioningControl
control.DirectoryEntry=com.aptrix.pluto.taxonomy.ProfileControl
```

Note: If the object has other values, insert a comma to separate this value from existing values.

3. Save the file.
4. Stop the WebSphere Application Server.
5. Restart the WebSphere Application Server so that the change can take effect.

The version control is automatically applied to all the new content created using a new Content Template, also created from V2.0. All contents migrated from V1.1 do not have this feature enabled.

To allow version control to all V1.1 contents, follow these next steps:

1. Open the browser and enter the following URL:

`http://[HOST]:9080/ilwwcm/connect/?MOD=VersioningEnablement`

2. Log on using the following user name and password (they are case-sensitive):
 - Specify User Name: Administrator
 - Specify Password: password

Once the Version Enablement Module is running, the log displays in the browser window. Objects from LWWCM V1.1 can now participate in version control. See the running results in Example 13-4.

Example 13-4 Module versioning enablement results

Log for Utility: Version Enablement Module

```
16:15:22.167 com.aptrix.pluto.util.PlutoUtilityModuleinfoStarting utility:
Version Enablement Module
16:15:26.413 com.aptrix.versioncontrol.VersioningEnablementModuleinfoEnabling
versioning for type: Content
16:17:13.307 com.aptrix.versioncontrol.VersioningEnablementModuleinfoEnabling
versioning for type: Style
16:17:22.39 com.aptrix.versioncontrol.VersioningEnablementModuleinfoEnabling
versioning for type: Cmpnt
16:18:20.103 com.aptrix.pluto.util.PlutoUtilityModuleinfoUtility completed.
```

Note: You can run this module (MOD=VersioningEnablement) as often as required. You need to run it every time you define version control for a new object in the `aptrixjpe.properties` file. After that, these objects will be able to participate in version control. Also, if the objects are created when the feature is not enabled and later, you want to enable them, run the same module.

Figure 13-12 on page 291 shows an example of an original V1.1 content without the version control utility (on the left) and the result of applying the Version Enablement module to it (on the right). Notice the Save Version button available on the second figure that is missing from the first figure.

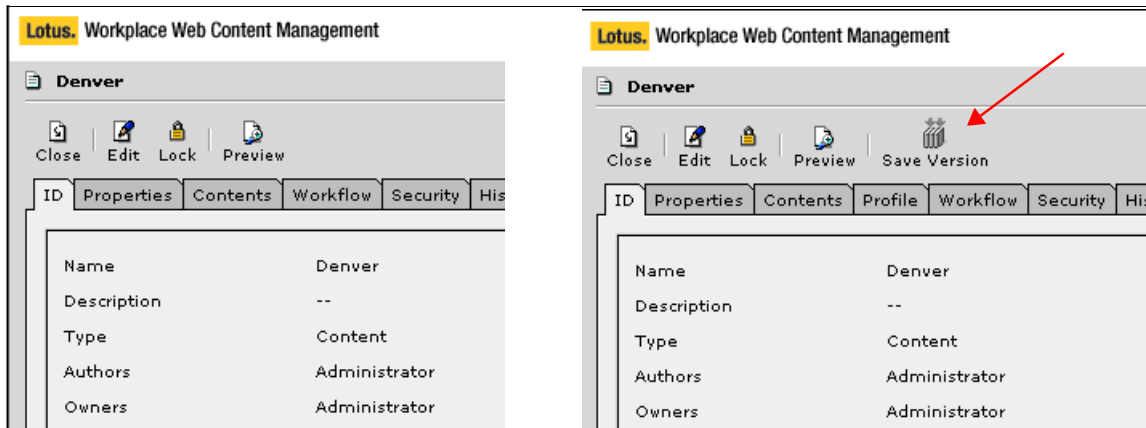


Figure 13-12 Migrated content before and after version enablement



Versioning

In this chapter, we introduce versioning, describe how to enable version control in LWWCM, how to store multiple versions of objects, and how to read, restore and delete objects from the LWWCM version library. Additionally, we address why and how to integrate the Content Manager versioning functionality from a Web site by using:

- ▶ Content Manager integrated versioning features
- ▶ Federated content component
- ▶ JSP component

14.1 Versioning overview

Content Manager and LWWCM support versioning. Versioning in the LWWCM publishing process with Content Manager as the back end content repository adds another spin to an integrated solution. In this section, we take a look at Content Manager and LWWCM versioning as stand alone products and as an integrated solution. We also examine how versioning of external Content Manager resources can be used in LWWCM Web site development.

LWWCM versioning is used to store multiple copies of an LWWCM object, keep a history of object development and restore previous versions of a current object. This is useful in a Web publishing process for component Web sites where the final Web sites are the result of several designer and content publisher meetings. It is also useful when preparing Web sites in advance where the final publishing decision depends on future events or company results.

Content Manager versioning allows versioning on item types, attributes and individual document parts. The maximum number of allowed versions can be limited to a predefined number per item type. For example, in our case study insurance scenarios, versioning enables us to keep track of the history of insurance contract offerings for digital tax reviews. Versioning can be set to keep track of the coherence of advertisement media files and the regional selling quarterly results. The attribute versions help to keep track metadata changes such as a policy holder's address and medical history.

With the integrated LWWCM and Content Manager solution, we need to consider how Content Manager versioning can be leveraged in the LWWCM Web publishing process.

14.2 Versioning of repository objects

Objects in LWWCM include Web site content and Web site building elements such as sites, site areas, and navigators. In order to keep a history of these objects and be able to restore snapshots, they can be refreshed in two ways:

- ▶ Versioning in LWWCM: Use the new LWWCM internal versioning features. Each version is a separate record in the underlying repository.
- ▶ Versioning in Content Manager: When you use the Content Manager as the back-end data repository, enable the automatic versioning of LWWCM objects in *Content Manager*. Every creation or update of objects in LWWCM using the LWWCM administration user interface creates a new version in Content Manager. One Content Manager record can contain many versions.

Important: Do not use the two approaches combined. This would lead to an undesirable outcome.

14.2.1 Versioning in LWWCM

Using this approach, the versioning functionality is only handled by the LWWCM framework. The relationship between the currently active object and its versions is stored in each object and is only interpretable by LWWCM. The back-end repository (in our case, Content Manager) itself does not know about the different versions of the object. Each single version of an object is a separate record in Content Manager which differs from active and other versioned records only by different attribute values.

Note: Versions in the LWWCM terminology always refer to non-live, previously-versioned objects. The live objects are not stored as versions. This potentially results in the current active objects being lost when they are replaced by their previous versions. To avoid losing the current active objects, always save them as new versions before restoring older versions.

Versioning features provided in LWWCM are:

- ▶ Batch versioning and batch labeling
- ▶ Batch restoration of versions based on labels or date
- ▶ Versioning a single object
- ▶ Restoration of single objects

By default, the following three objects are version-enabled in LWWCM:

- ▶ Content
- ▶ Style
- ▶ Components

To enable other object types, refer to “Enabling versioning” on page 298.

Important: Only versions of individual objects can be saved. Any hierarchical relationships to other objects are not saved. For example, if you save a version of a site area that has child site areas and you delete the child site areas, the child site areas will not be restored if you restore the parent site area.

How versions are physically stored in Content Manager

This section discusses in depth how versions are physically stored in Content Manager. If you are only interested in how to use versions, skip this section.

If LWWCM uses Content Manager as a back-end repository server, the first time you save a version of a version-enabled object, two new objects will be created in Content Manager:

► Version

This is a copy of the object you just versioned, wrapped up in additional information such as the version number, when it was versioned, logging information and a reference to its parent object.

► Version catalog

This is a catalog of all available versions regarding one object. The catalog contains metadata information about each available version such as its version number, creation time, and the reference to the actual version object. The Version Library view in the LWWCM administration user interface is based on the information stored in objects like the version catalogue.

Any additional version of the same object would create a new version object and update the catalog accordingly.

Important: Do not delete records in Content Manager manually. This would lead to inconsistencies in LWWCM. If any consistency problems occur, stop the LWWCM server, delete the indexes then restart the LWWCM server.

Figure 14-1 on page 297 shows the version catalog, the versions and the actual content, Life-term Insurance, as they are stored in Content Manager.

To get this view, do the following:

1. Launch the Content Manager Windows Client.
2. Perform a search on item type *AJPEData* where:
 - *JPE* Server ID is the server ID of the LWWCM server (for example, 1).
 - Name is *Life-term**. Do not forget the * to retrieve all objects starting with this name.

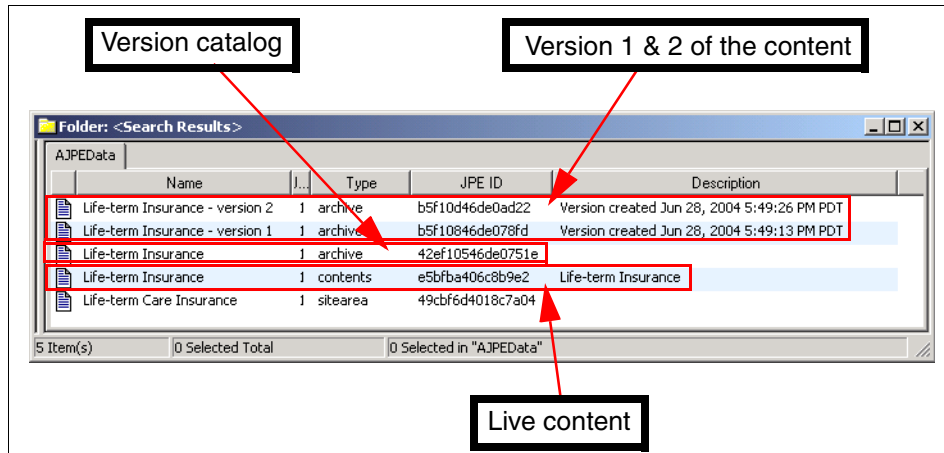


Figure 14-1 Different LWWCM versions in Content Manager

Figure 14-2 shows the architecture of the version feature in LWWCM. The content shown on a Web page is always derived from the published object, the live object. The published object (contents and site area) is stored separately from the version objects (type archive) and the version catalog (type archive).

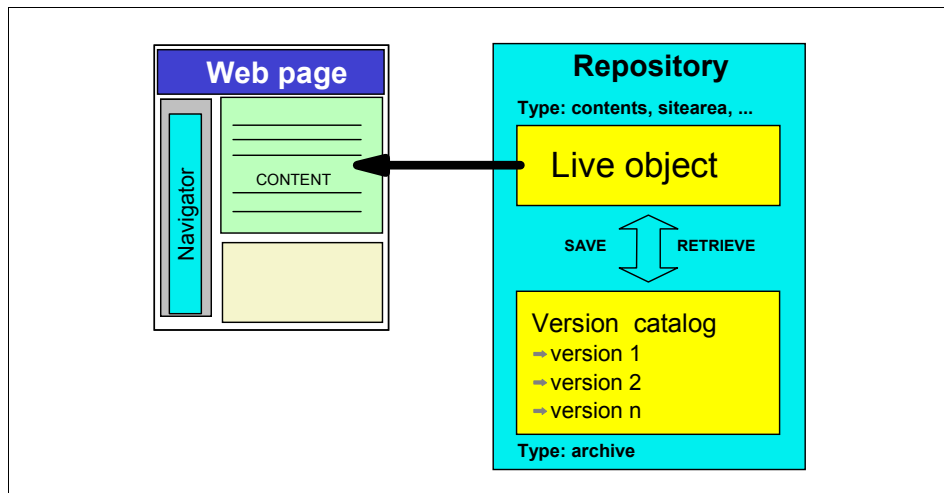


Figure 14-2 Versioning

Enabling versioning

The `aptrixjpe.properties` file can be edited to enable the version control of different objects. To enable version control, add the following to any of the `control.[object]` settings:

```
com.aptrix.versioncontrol.VersioningControl
```

Multiple entries must be separated by a comma.

Tip: The `aptrixjpe.properties` file can be found in:

```
[LWWCM_HOME]\config\aptrixjpe.properties
```

The following is an example of the steps required to enable content, component and page style objects for versioning in LWWCM:

1. Open the properties file and locate the following section in the file:

```
#control properties
```

2. Adjust the `control.Content`, the `control.Cmpnt` and the `control.Style` lines. Note that the content control is also enabled for workflow and profile control:

```
#control properties
```

```
.....
```

```
control.Content=com.aptrix.versioncontrol.VersioningControl,com.aptrix.pluto.workflow.WorkflowControl,com.aptrix.pluto.taxonomy.ProfileControl
```

```
control.Style=com.aptrix.versioncontrol.VersioningControl
```

```
control.Cmpnt=com.aptrix.versioncontrol.VersioningControl
```

```
.....
```

3. Restart the LWWCM server.

New objects of type content, style or `cmpnt` are now version-enabled. Existing objects of these types will not be version-enabled by default but can be subsequently version-enabled by executing the following URL:

```
http://\[HOST\]:\[PORT\]/\[CONTEXT\_ROOT\]/connect/?MOD=VersioningEnablement
```

Using LWWCM versioning

Once versioning is enabled for objects, we can use the LWWCM versioning feature to:

- ▶ Create new versions for the objects.
- ▶ Create a snap-shot of all version-enabled objects.
- ▶ Restore versions of the objects.

Creating a new version

You can create a new version by using the **Save Version** button when working with an object.

Note: The Save Version button is only available in read mode.

We show you how to create a new version of an object using our case study content object, Life-term insurance, as an example:

1. Launch the LWWCM administration user interface.
2. Expand **Content Management** -> **Content Library**.
3. From the right-hand panel, select the **Life-term Insurance** content.
4. Click **Read** to open the content in read mode.
5. Click **Save Version** to trigger the creation of a new version.
6. Enter one or more labels, separated by semicolons, to identify the new version. See Figure 14-3.
7. Click **OK**.

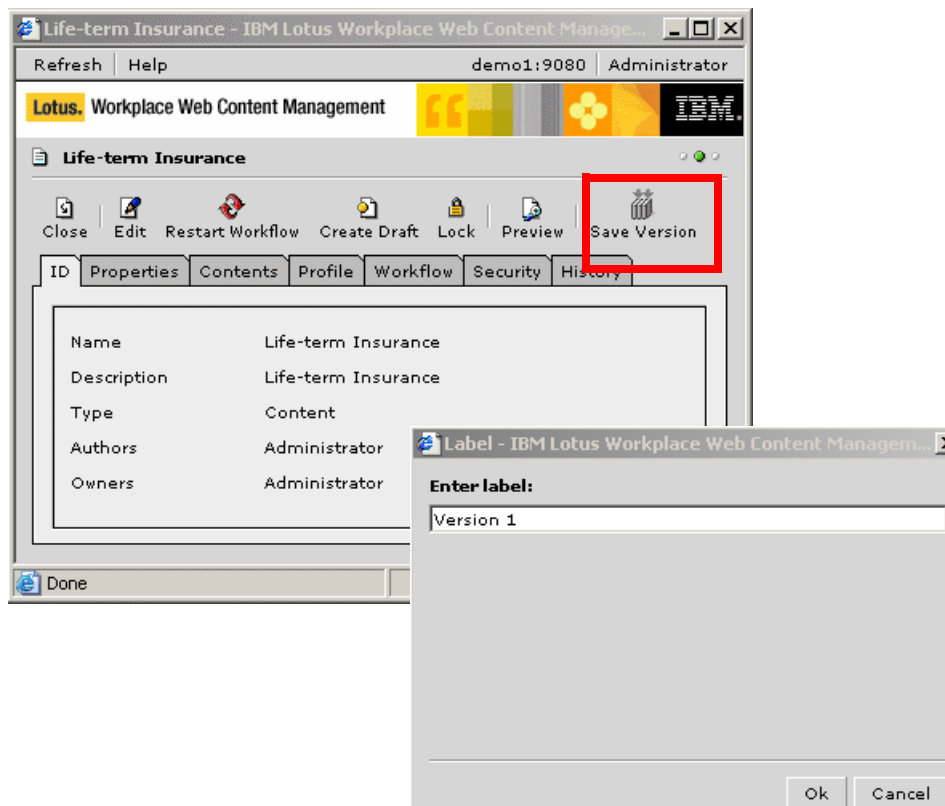


Figure 14-3 Save version of a content object

Creating a snap-shot of all objects

A common need is to create a snap shot of all available object states at a certain point in time so you can revert to this state when needed.

Important: Only objects which are published and version-enabled in `aptrixjpe.properties` are affected by this action.

To do this task, do the following:

1. Launch the LWWCM administration user interface.
2. Expand **Version Management -> Version Library**.
3. Click **Version All**. This will create the first or an additional version of all version-enabled objects.
4. Enter one or more labels, separated by semicolons, to identify the new version. This label will later be used to restore all objects with this label.
5. Creating versions for all version-enabled objects may take some time. Wait until the creation is finished. To check if all the creations are done, click **Check Status**. The message box will tell you about the current process and its progress (Figure 14-4).

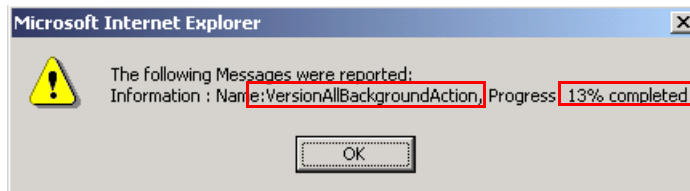


Figure 14-4 Version all process status

You can assign additional labels to the latest version stored for each object. Follow the steps below:

1. Click **Label All**. This will label all the latest versions in version-enabled objects with a specific label.
2. Enter one or more labels, separated by semicolons, to identify the new version. You can use this label later to restore all objects assigned to it.
3. Wait until the labeling is finished.

Restoring versions

You can restore all objects with a specific label, or you can restore a version for just one object.

When restoring a single object to a particular version, you can:

- ▶ Define the restored version as a draft of the currently active object.
- ▶ Have the restored version overwrite the currently active one.

Note: When you restore all objects to a specific version, the currently active objects will be overwritten with the stored object versions unless you save the new ones first. We highly recommend you save the current active version prior to restoring a previous version.

To restore all objects to a particular version, do the following:

1. Launch the LWWCM administration user interface.
2. Expand **Version Management -> Version Library**.
3. Click **Restore All**.
4. Enter the label of the versions you want to restore or a date value.
5. Click **OK**.
6. Wait until all objects are restored.

To restore a single object to a particular version, do the following:

1. Launch the LWWCM administration user interface.
2. Expand **Content Management -> Content Library**.
3. Expand the object type of the object version you want to restore.
4. Mark the concerned object and click **Show Versions**.
5. Select the version you want to restore. Click **Restore** to overwrite the currently active object or click **Restore as draft** to restore the selected version as a draft of the currently active object (see Figure 14-5).
6. Click **OK** to confirm the selection.

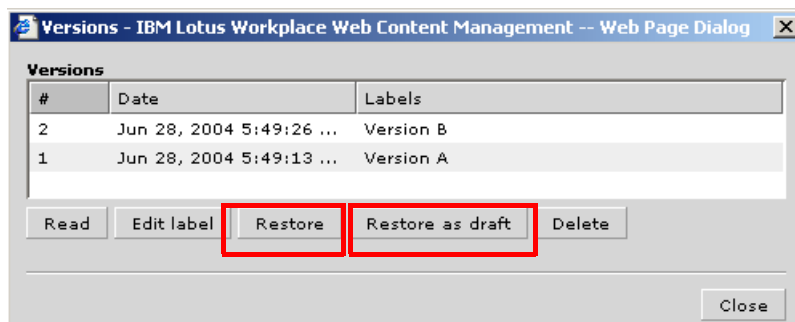


Figure 14-5 Version management in LWWCM

14.2.2 Versioning in Content Manager

If you use Content Manager as the repository type, enable the automatic versioning of LWWCM objects in Content Manager. Using this approach, the Content Manager performs versioning tasks. The relationship between the currently active object and its previous versions is stored inside Content Manager. Every time you create or update an object in LWWCM using the LWWCM administration user interface, Content Manager creates a new version of the corresponding AJPEData object. In LWWCM versioning, different versions of an object in LWWCM are stored as Content Manager version records and not as separate AJPEData records.

Figure 14-6 shows the difference between the internal storage in Content Manager for using the LWWCM versioning (the top screen) and using the Content Manager versioning feature.

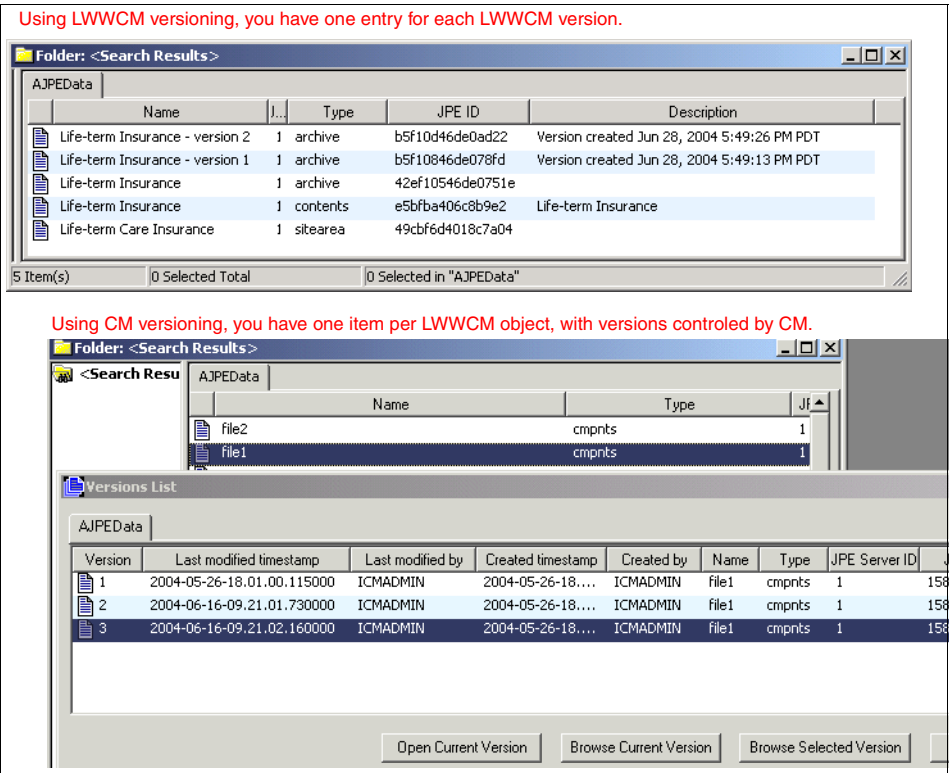


Figure 14-6 Comparison of LWWCM and Content Manager versioning

From a Content Manager Windows Client or eClient's AJPEData search result list, select an AJPEData object and open the version list to view the existing Content Manager versions. See the second portion of Figure 14-6. Note the

LWWCM file and image resource components are stored as AJPEResources objects in Content Manager. For more information about LWWCM objects stored as AJPEData and AJPEResources objects in Content Manager, see section 6.4, “How LWWCM objects are stored in CM” on page 108.

To enable Content Manager versioning for all LWWCM objects, do the following:

1. Open the `aptrixjpe.properties` file located in the `[LWWCM_HOME]\config` directory.
2. Locate the #Content Manager persistency settings section and change the following line:

```
cm.versioning=true
```

3. Restart the LWWCM server.

New versions will be created in Content Manager for the AJPEData item type objects whenever an LWWCM object is updated in the LWWCM framework. For the AJPEResources objects, there is only one version stored in Content Manager.

Note: At the time of writing, we do not recommend using the Content Manager versioning feature in LWWCM V2.0 for the following reasons:

- ▶ LWWCM is not aware of any way to restore objects that are versioned within Content Manager. Unless you use Information Integrator for Content APIs and manually program to get the previous versions, using Content Manager versioning feature with LWWCM might not be helpful.
- ▶ If you use the Content Manager versioning feature and then syndicate the LWWCM server to another server, the Content Manager versioning information will not syndicate to this new server since LWWCM is not aware of its existence.
- ▶ The deletions in Content Manager are not handled correctly in this release. When an object is deleted from the LWWCM framework, only the first Content Manager version of the object is removed; all other versions stay orphaned in the system. This, however, is completely transparent to the LWWCM users.

14.3 Versioning external resources

The only way to version external resources is to use Content Manager versioning.

With LWWCM versioning, copies of links to external resources within an LWWCM object, such as Content Manager resources, are not kept in the version

library. If you delete the external resource, you cannot use the LWWCM version library to restore the resource. LWWCM does not provide a front end to retrieve previous versions of external resources. The retrieval of previous versions would have to involve some manipulation.

14.3.1 Versioning in federated, JSP components and connect tags

This section explains other types of versioning for specific purposes. For more information about these, follow the provided links.

Using the *federated content component* (federated component) you can render a URL, attributes or the actual content of external resources into your Web page. Additionally, the component gives you the ability to select the Content Manager version of the content you want to retrieve. For information about the federated content component, refer to 7.3, “Federated content component” on page 123.

With the *JSP component*, you have the ability to define which Content Manager version of the records you want to access. To perform this task, you enhance the search string with version criteria to limit the result set to a specific (mostly the latest) version. For more information about the JSP component, refer to 10.2.1, “Leveraging the CM/II for Content Java APIs” on page 220.

For information about accessing different Content Manager item type parts and attribute versions with the search connect tags syntax, see also the W3C standard for XPath/XQuery-based XML document management.

The following example shows how to use a search tag to get the latest version for the InsuranceVideo item type:

```
'/InsuranceVideo[@VERSIONID = latest-version(.)]'
```

This query finds all the videos 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. For more details about the connect tag using Content Manager's XPath syntax, refer to *IBM DB2 Content Manager for Multiplatforms - Workstation Application Programming Guide Version 8.2*, SC27-1347, starting on page 184, “Example searches using the query language.”

14.3.2 Versioning using Content Manager

This section describes briefly how to use versioning features in a pure Content Manager environment. As mentioned in “14.3.1, “Versioning in federated, JSP components and connect tags” on page 304, the Content Manager versioning possibility can be integrated in the LWWCM Web content by using a federated content component, JSP component, or with search connect tag syntax.

Content Manager has a versioning feature for item objects and attributes. For example, in our case study, if you change the IT_Group attribute for a specific item object to another IT_Group name, then Content Manager creates a new, updated version of the item. The version policy that you set on the Definition page of an item type applies also to the attribute values. For our InsuranceVideo item type definition, as shown in Figure 14-7 on page 306, we allow a maximum number of seven versions. When you change InsuranceVideo, a new version is created. If you exceed the maximum number of versions, the oldest version is automatically deleted when saving the next new InsuranceVideo object version.

The following version policies are possible:

- ▶ Never create: A stored item is updated every time. Only one version of this item exists.
- ▶ Always create: Always create a new version of the item whenever it is updated.
- ▶ Prompt to create: Users decide whether they want a new version to be created when they are updating an item.

For the last two policy types, Always create or Prompt to create, the number of versions can be set to a maximum number as in our sample in Figure 14-7 on page 306, or it can be set to allow an unlimited number of versions.

Item Type Properties - InsuranceVideo

Auto-linking	Foreign Keys	Logging	Document Management	User Exits
Definition	Access Control	Attributes	Default Storage	

Name: * InsuranceVideo

Display name: * InsuranceVideo [Translate...](#)

New version policy:

- ☐ Never create
- ☐ Always create
- ☒ Prompt to create

Maximum total versions:

- ☐ Unlimited
- ☒ Limited to 7

Item type classification: Document

Media Object (XDO) Class:

Text Search:

- ☒ Text searchable [Options...](#)

Item retention period:

- ☒ Forever
- ☐ For 1 Years

Start item on Process: Priority: 0

OK Cancel Apply Help

Figure 14-7 Set the versioning policy for the InsuranceVideo item type

If the item type that you are creating is classified as a resource item or document part, the version policy from the definition window 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 on the Document Management window, accessible by clicking **Add** in the Document Management tab (see also Figure 14-8 on page 307).

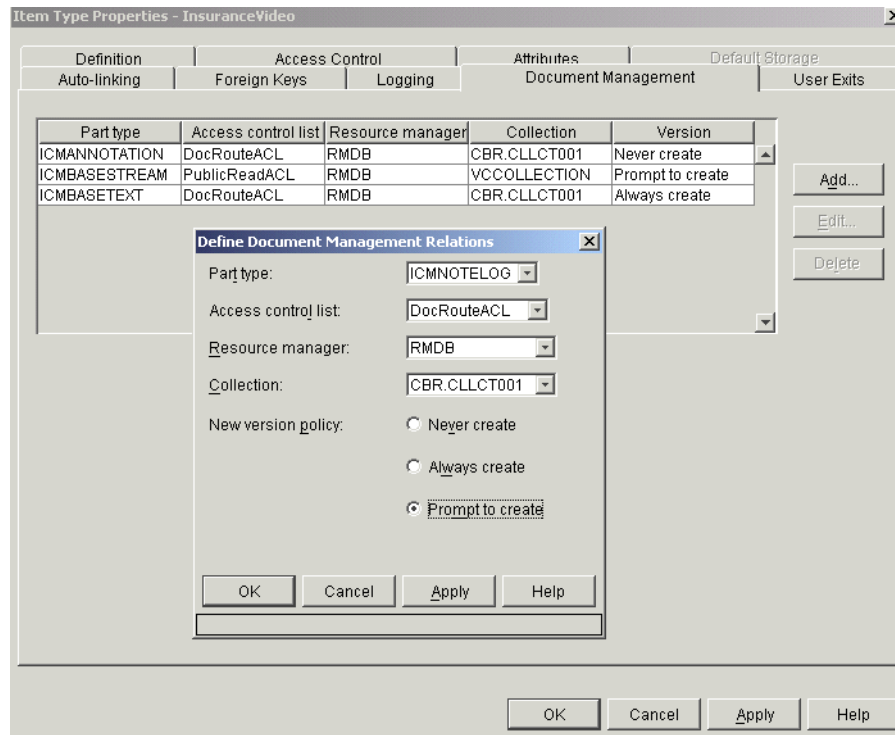


Figure 14-8 Supplemental version policy for specific document parts



A

Installation supplement

This appendix provides the following information:

- ▶ Accessing a remote LS database
- ▶ Cleaning up a Windows system prior to installation

Accessing a remote LS database

Access remote LS DB in a DB2 Command Window (or use the DB2 Command Center) to first catalog the remote TCP/IP node and then the remote LS DB:

```
CATALOG [ADMIN] TCP/IP NODE node-name REMOTE hostname [SERVER service-name]
[SECURITY {SOCKS}] [REMOTE_INSTANCE instance-name]
[SYSTEM system-name] [OSTYPE os-type] [WITH "comment string"]
```

Note: SERVER service-name is mandatory for CATALOG TCP/IP NODE
and not allowed for CATALOG ADMIN TCP/IP NODE

```
CATALOG DATABASE database-name [AS alias] [ON drive | AT NODE node-name]
[AUTHENTICATION {SERVER | CLIENT | DCS | DCE SERVER PRINCIPAL principalname
|
KERBEROS TARGET PRINCIPAL principalname | SERVER_ENCRYPT | DCS_ENCRYPT}]
[WITH "comment-string"]
```

In our setup, we use the following commands to access a remote LS DBs on a Windows machine demo2 and on a AIX machine named zaire:

- ▶ catalog tcpip node <node> remote <hostname> server <port>
 - catalog tcpip node DEMO2 remote demo2.svl.ibm.com server 50000
 - catalog tcpip node zaireaix remote zaire.svl.ibm.com server 60002
- ▶ catalog database <remote db name> as <alias> at node <node name>
 - catalog database ICMNLSDB as DEMO2LDB at node DEMO2
 - catalog database ICMNLSDB as ZDB at node zaireaix

For the node name, choose whatever name you want (no more than eight characters long). To find the right port number for an remote LS AIX installation, check the remote system in /usr/etc file services, and type `cat services` to find the connection port number for the database instance for this remote LS database. On Windows, go to the file C:\WINNT\system32\drivers\etc and enter type services.

Also check the DB2COMM configuration parameter with **db2set**, in a DB2 command window, if you use TCP/IP communication, to see if it is configured:

```
DB2COMM=TCPIP, NETBIOS
```

If you are not familiar with DB2 commands, just use the DB2 Control Center to access a remote system and to catalog a remote DB2 database.

After the test, enter a **db2 connect** command from the DB2 command window; for our sample, we try a connection against the remote AIX database ICMNLSDB (alias ZDB):

```
db2 connect to ZDB user ICMADMIN using <ICMADMINs password on zaire>
```

Note: If you get an SQL error with SQLSTATE=08001 see 5.4, “Problem determination and troubleshooting” on page 78 in the 5.4.3, “DB2 and NSE” on page 80 subchapter.

Cleaning up a Windows system prior to installation

Always try to install on *clean systems* for the ILWWMC/CM server installation, to avoid a situation where troubleshooting becomes necessary. For performance and maintenance reasons, this kind of approach also makes sense.

Below, we describe how you can guarantee that you install the ILWWCM or CM servers on a clean Windows system for a new ILWWCM/CM system setup:

1. Remove previous versions if they are no longer needed by clicking **Start -> Settings -> Control Panel -> Add/Remove Programs**. Always reboot after the full uninstallation.
2. From the Command Prompt window, enter **regedit** then go to **My Computer -> HKEY_LOCAL_MACHINE -> Software** and remove any existing entries where the programs were already removed. Delete any corresponding Services entry for this programs by clicking **My Computer -> HKEY_LOCAL_MACHINE -> System -> ControlSet00.. -> Services** and **My Computer -> HKEY_LOCAL_MACHINE -> System -> CurrentControlSet -> Services**.
3. Delete any leftover subdirectories from previous installations

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 314. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Content Manager Implementation and Migration Cookbook*, SG24-7051
- ▶ *Lotus Workplace Web Content Management*, SG24-6309

Other publications

These publications are also relevant as further information sources:

- ▶ *Content Manager for Multiplatforms - Planning and Installing Your Content Management System*, GC27-1332
- ▶ *Content Manager for Multiplatforms / Information Integrator for Content - Workstation Application Programming Guide Version 8.2*, SC27-1347
- ▶ *Net Search Extender Administration and User's Guide*, SH12-6740

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ Content Manager support home page:
<http://www.ibm.com/software/data/cm/support.html>
- ▶ Information Integrator for Content (EIP) support home page:
<http://www.ibm.com/software/data/eip/support.html>
- ▶ DES home page:
<http://www.lotus.com/products/des.nsf/wdocs/home>

- ▶ WebSphere Portal and Lotus Workplace catalog:
<http://catalog.lotus.com/wps/portal/portalworkplace>
- ▶ WebSphere Portal Information Center home page:
<http://publib.boulder.ibm.com/pvc/wp/502/ent/en/InfoCenter/index.html>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

tag 152

Numerics

08001 82

10061 82

A

access

Content Manager content 189

Access Control List 25

access control list 26

access levels

LWWCM 238

access module 64

access.log 64

ACL 25, 249

admin_error.log 64

administration user interface

LWWCM 7

advantages

CM compared to JDBC repository for LWWCM 39

integration 171

Using CM as external resources 38

Using CM as repository 37

agents 82

AJPEData 108, 296, 302

attributes 111

storing 109

viewing item type 109

viewing items 110

XMLContent data 113

AJPEGUI module 108

AJPEResources 108, 303

attributes 112

storing 109

viewing item type 109

viewing items 110

alias name

database 89

APIs

Information Integrator for Content 22

approval

content 5

approver 18

aptrixjpe.properties 94, 100–101, 200, 298

architecture

Content Manager 22

Information Integrator for Content 19

LWWCM 5, 13

Resource Manager 26

WebSphere Portal Server 188

archive 20

archiving

computer output 20

attributes

creating in Content Manager 119

authentication 24, 249

authenticity 10

author 5, 49

content 4

authorization 24, 249

authorization flow 10

auto-linking 26

automated notification 18

AV 20

B

backup 20

Body tag 54

Boolean search 21

buffer pool 82

bulk document loading 18

business processes 5

C

C++ APIs 22

C2A 190

cache LWWCM components

using connect tags 146

caching

content 5

caching services 4

case study 42

setup 70

- catalog 98
- Category Management
 - Site Management 7
- category taxonomy 7
- check
 - installed software version 71
- checklist
 - validation 72, 75
- classification
 - content 7
 - item type 25
- client application 22
 - Working with Library Server and Resource Manager 31
- Client for Windows 22, 30
- cm81install.log 64
- cmbds.ini 103, 105, 107, 124
- cmbscmrsvs.ini 100, 103, 107
- cmlevel.exe 71
- COLD 20
- collaboration
 - team 5
- CommonStore 17, 20
- comparison
 - Domino and Java editions 7
- component design 13
- Component Library 10, 116
- component page design 8
- component tags
 - using within page design 133
- compound document 18
- computer output to laser disk 20
- configuration files 64, 97
- configure the log level
 - LWWCM 87
- connect tags 12, 58, 116, 139, 304
 - display data from external resources 141
 - display data from Web 143
 - security 252
 - using within page design 141
- connect.cfg 124, 164
- connect.log 164
- connector 19, 117
- content
 - author 4
 - component 10
 - creating 56–57
 - creating in Content Manager 122
 - creation tool 10
 - delivery 189
 - design 5
 - dynamic 12
 - referencing into site area 57
- Content Library 10
- Content Management
 - LWWCM module 7, 10
- Content Manager 16
 - architecture overview 22
 - creating attributes 119
 - creating content 122
 - creating item type 120
 - integration with VideoCharger 259
 - product portfolio 17
 - publishing content 115
 - repository for external sources 169
 - repository for LWWCM 92, 169
 - retrieving data 143
 - search possibilities 210
 - software requirements 68
 - Using as external resources advantages 38
 - Using as repository for LWWCM advantages 37
 - validation checklist 75
 - versioning 302, 304
 - Working with LWWCM 33
- Content Manager content
 - access using portlets 189
- Content Manager Information Integrator for Content 18
- Content Manager Installation log files 64
- Content Manager OnDemand 17, 20
- Content Manager portlets 189, 201
- Content Manager security 248
- Content Manager V8 connector 103
- Content Manager versioning 294
- content portlet 195
- content repository
 - migrating 278
- Content Template 10
- content template 8, 56
 - creating 50
- content templates 49
- content viewer portlets 189
- contributor 18
- create
 - Web sites 9
- creating
 - attributes in Content Manager 119
 - content 56–57

- content in Content Manager 122
- content template 50
- federated component in component library 129
- federated component within a content object 131
- federated content component 129
- federated entity 127
- federated search template 128
- item type in Content Manager 120
- JSP component 153
- JSP component in component library 153
- JSP component within a content object 153
- page layout 55
- portal page 199
- site area 44
- snap shot of all objects 300
- subscriber 273
- syndicator 273
- video component in LWWCM 268
- workflow 48
- creation 13
- creator 18

D

- data modeling, 24
- data storage 101
- database
 - alias name 89
- database creation
 - Library Server and Resource Manager 64
- database query language 23
- database.log 64
- DB2
 - verify installation 74
- DB2 log file 81
- DB2 Runtime Client 168
- DB2 trace utility 81
- db2cmd 81
- db2diag.log 81
- db2level 71
- db2rbind 82
- db2text 75
- db2trc 80
- db2trc.exe 81
- debug.log 164
- DefaultEncoding value 88
- deleting indexes 96
- delivery 13

- DES 234
- design
 - content 5
 - Web page 6, 9
- Design & Development
 - LWWCM module 7–8
- design component 10
- designer
 - Web page 4–5
- developer 49
- development stage 12
- development workstation
 - Information Integrator for Content 117, 168
- DGL0394A 96
- DGL0587A 96
- Digital Asset Management 16
- digital rights management 21
- Directory Access Protocol 24
- disadvantages
 - integration 171
- display
 - integration level 191
- display data from Web
 - using connect tags 143
- distance learning 16
- distribute content 12
- DNS 65
- document 26
 - item type classification 26
- document creation 18
- document life cycle management 18
- document loading
 - bulk 18
- document management 5
- Document Manager 16, 18
- document part 25
- document routing 24, 26
- document search 21
- DOMEA 16
- Domino edition
 - LWWCM 6
- Domino Extended Search 234
- draft 11
- DRM 21
- dynamic
 - content 12
- dynamic content portlet 194

E

- eClient 22, 30–31
- ECM 16
- e-government 18
- Electronic Media Management System 17, 21
- e-mail 18, 20
- embedded video 269
- EMMS 17, 21
- enable versioning 298
- end-to-end solution 4
- Enterprise Content Management 16
- Enterprise Resource Planning 16
- ERP 16
- error.log 64, 87, 164
- expiring
 - Web pages 4
- external databases
 - display using connect tags 141
- external resources
 - versioning 303
- external sources
 - retrieve data 116

F

- features
 - differences in Domino and Java editions 6
- FED connector 103
 - testing 107
- federated attributes
 - rendering 134
- federated component
 - creating in component library 129
 - creating within a content object 131
- federated component form 130
- federated connector 103
- federated content
 - rendering 136
- federated content component 12, 58, 116, 123, 304
 - creating 129
 - security 252
 - setting up 124
- federated content links
 - rendering 135
- federated data source 12
- federated entities 125
- federated entity
 - creating 127
- federated search 215–216, 219

- federated search template 101, 123
 - creating 128
- FederatedCmpnt tag 133–134
- FederatedLibCmpnt tag 133
- FederatedProxy module 135
- file plan 18
- file system 27
- fixed disk 20
- folder 26
- foreign key 26
- free text search 21
- full text search 23, 211
- fuzzy search 21

G

- G723 20
- government 18
- GPP 21
- groups
 - default 240

H

- hints and tips 78, 80
- HTML 21

I

- IA 9
- IBM Directory Server 24
- IBM HTTP Server
 - logs 64
 - verify installation 72
- ICM connector 103
 - testing 107
- ICMAnnotations 25
- ICMBASE 212
- ICMBase 25–26
- ICMBASESTREAM 262
- ICMBaseStream 25
- ICMBASETEXT 212
- ICMBaseText 25–26
- ICMClientLog.ini 65
- icmcrsdb.log 64
- icmcrrmdb.log 64
- ICMfetchContent 81
- ICMfetchFilter 81
- ICMNotelog 25
- icmserver.log 64

- ICMStream 26
- iFrame 231
- iFrames 220
- import videos 262
- importing
 - text searchable items 214
- indexes 89
 - deleting 96
- Information Architecture 9
- Information Integration for Content
 - validation checklist 75
- Information Integrator for Content 16, 18
 - APIs 22
 - architecture 19
 - development workstation 117
- Information Integrator for Content portlets 204
- installation
 - verify 72, 74
 - VideoCharger 258
- instant message 18
- integrate
 - why and when 35
- integrated
 - integration level 191
- integration 168
 - advantages and disadvantages 171
 - external resources 37
 - maintenance 171
 - security considerations 171
 - VideoCharger and Content Manager 259
- integration level 191
- interactive integration
 - between people and content 10
- internal LWWCM search engine 219
- intuitive click to action navigation 190
- IT JumpStart intranet Web site 42
- item type
 - creating in Content Manager 120
- item type classification 25

J

- Java APIs 22
- Java edition
 - LWWCM 6
- Java heap size 93
- JDBC driver 79
 - classpath 93
- JPE Server ID 95, 296

- JSP code
 - sample, connection to CM 156
 - sample, simple search 158
- JSP component 58, 116, 151, 304
 - configuration 154
 - creating 153
 - creating in component library 153
 - creating within a content object 153
 - security 253
 - setting up 152
 - using 155
- JSP components 13
- JSP program
 - execute outside of LWWCM 165
- JVM 92

L

- LBR 20
- LDAP 24, 190
- LDAP directory
 - retrieving data 142
- Library Server 22, 24
 - database creation 64
 - remote access 97
 - Working with Resource Manager and client application 31
- Library Server log file 64
- Lightweight Directory Access Protocol 24
- link
 - integration level 191
- live 11
- locklist 82
- log files 64, 164
- log level
 - configure (LWWCM) 87
- logging level 26
- long term storage 20
- look and feel 4, 8, 12
- Lotus Domino 20
- Lotus Domino Directory Notes Address Book 24
- Lotus Workplace Web Content Management 16–17
- LWWCM 16–17, 32, 73
 - architecture 5, 13
 - Domino and Java editions 6
 - initial setup 96
 - integrating in WPS 190
 - search module 219

- software requirements 67
- verify installation 74
- Working with Content Manager 33
- LWWCM log files 64
- LWWCM modules 7
- LWWCM objects
 - storing in Content Manager 108
- LWWCM portlets 193
- LWWCM search engine 219
- LWWCM security 238
- LWWCM server routing information 99
- LWWCM user interface 7
- LWWCM version control
 - enabling 289
- LWWCM versioning 294
 - using 298

M

- maintenance
 - integration 171
- management 13
- maxagents 82
- maxappls 82
- media archive 27
- menu portlet 198
- messaging 5
- Microsoft Active Directory 24
- Microsoft Exchange 20
- migrated
 - integration level 191
- migrating
 - content 12
 - content repositories 278
- migration policy 27
- migrator 28
- MJPEG 20
- mobile Sales Force Automation 16
- modules
 - LWWCM 7
- MPEG 20
- MPEG1 20
- MPEG2 20

N

- navigation portlet 194
- navigator 7
 - Web site 43
- Net Search Extender 16, 21–22, 219

- netstat 65
- network consideration
 - LWWCM and CM installation 65
- nodes 82
- NSE 16, 21, 212, 219
 - verify installation 74
- NSESearch.jsp 222

O

- OnDemand 17, 20
- one-way transportation 272
- optical jukebox 20

P

- page design 8–10, 13, 53, 56
 - create 9
 - using component tags 133
 - using connect tags 141
- page layout
 - creating 55
- pages design 8
- paper-based process 18
- parametric search 210
- personalization 4
- phrase search 21
- play videos 262
- plotting 18
- port
 - setting different port 65
- port conflict 65
- portal page 193
 - creating 199
- portal page layout
 - sample 193
- portal search engine 234
- portlet 190
- portlet pattern
 - advantages and disadvantages 191
- portlets 190
 - types 193
- predefined document parts 25
- privilege 24, 249
- privilege group 249
- privilege set 25, 249
- problem determination 78
- process management 18
- product portfolio
 - Content Manager 17

- production stage 12
- proximity search 21
- PSE 234
- publish
 - documents 4
- publishing
 - Content Manager content 115
 - Web pages 4
- PublicReadACL 262
- purger 28

Q

- QuickTime 20

R

- ranking
 - search 21
- record life cycle 18
- record management standards 16
- Record Manager 18
- record structure 18
- Records Manager 17
- Redbooks Web site 314
 - Contact us xviii
- remote client workstation 23
- remote Library Server database 97
- remote printing 18
- RendererModule 147
- rendering
 - federated attributes 134
 - federated content 136
 - federated content links 135
- rendition service 18
- reorgchk 82
- replicator 28
- repositories
 - migrating 278
- repository 18
 - using CM for external sources 169
 - using CM for LWWCM 169
 - Web page 5
- resource
 - item type classification 25
- Resource Manager 16, 20, 22, 26
 - architecture 26
 - configure 260
 - database creation 64
 - manage storage of objects 28

- Working with Library Server and client application 31
- restoring versions 300
- retrieve data
 - from external sources 116
- retrieving complete page
 - from Web site 144
- retrieving data
 - from Content Manager 143
 - from different source 13
 - from external sources 58
 - from LDAP directory 142
 - SQL database 142
- retrieving partial page
 - from Web site 144
- reviewer 18
- runstats 82

S

- Sales Force Automation 16
- sample
 - TSearch query 218
- sample code
 - using wcm JSP tags 160
- sample JSP code
 - connection to CM 156
 - simple search 158
- SAP R/3 20
- search 21
 - integrating 219
- search approaches 234
- search engine
 - LWWCM 219
- search module
 - LWWCM 219
- search possibilities
 - Content Manager 210
- search template 89, 125
 - creating 128
 - federated 101, 123
- search videos 262
- security 200
 - connect tags 252
 - Content Manager 248
 - federated content component 252
 - integration 171
 - JSP component 253
- Security & Workflow

- LWWCM module 7
- security sections
 - LWWCM 238
- security setting 4
- Server Configuration Utility 106
- server operating system
 - consideration 169
- server routing information
 - LWWCM 99
- SFA 16
- shared resources 116
- Single Sign On 190
- site area 8, 56–57
 - create 9
 - creating 44
- Site Framework
 - Site Management 7
- Site Management
 - LWWCM module 7
- smitty 71
- SMS 26, 260
- snap-shot
 - creating 300
- socket error 66, 82
- software requirements
 - Content Manager 68
 - LWWCM 67
- software versions
 - check 71
- sort heap 82
- SP 22–23
- space management facility 20
- SQL database
 - retrieving data 142
- stager 28
- staging
 - content 5
- static content portlet 194
- storage 13
 - long term 20
- storage managed system 260
- Storage Manager 17
- storage system 26–27
- stored procedures 22–23
- streaming 20
- structured document search 21
- subscriber 8, 88, 272
 - creating 273
- syndication 4, 12, 272

- setting up 272
- Syndication & Deployment
 - LWWCM module 7–8
- syndicator 8, 88
 - creating 273
- System Administration Client 29
 - interface 30
- System Administration Client log file 64
- system Administration Client, 29
- System Administrator Client 22
- system architecture 167

T

- tape drive 20
- taxonomy
 - category 7
- TConnect 165
- TConnect.java 107
- TCP/IP
 - address 65
 - error 82
- team collaboration 5
- templates
 - pre-build 4
- text metadata 23
- text search 22, 212
- text search options 213
- text searchable items
 - importing 214
- thesaurus search 21
- Tivoli Storage Manager 20, 27
- trace levels 164
- transportation
 - syndication 272
- troubleshooting 78, 96, 164, 286
- TSearch query 218
- TSM 20, 27
- two-way transportation 272

U

- UDF 81
- UK PRO2 16
- update
 - Web sites 9
- upgrading
 - LWWCM 287
- US 5015.2 16
- user 125

- User Defined Function 81
- user interface
 - LWWCM 7
- user management 200
- users
 - default 240
- using LWWCM versioning 298
- UTF8 88, 92

V

- validation checklist 72
- VERS 16
- version control 89
 - LWWCM, enabling 289
- Version Library 7
- Version Management
 - LWWCM module 7
- versioning 4, 11, 26, 293
 - Content Manager 294, 302
 - enabling 298
 - external resources 303
 - federated, JSP components and connect tags 304
 - LWWCM 294
 - LWWCM, using 298
 - using Content Manager 304
- versions
 - restoring 300
- Victorian Electronic Record Standard 16
- video
 - incorporating within a Web page 269
- video component
 - creating in LWWCM 268
- video links 269
- VideoCharger 16, 20, 27, 255
 - installation 257–258
 - integration with Content Manager 259
- VideoCharger log files 65
- VideoCharger player
 - launch 265
- videos
 - import, search, and play 262
 - integrate to Web site 267

W

- WAV 20
- wcm JSP tag 160
- Web Content Management 16

- Web page 9
- Web page design 6, 9
- Web page designer 4
- Web site
 - integrate Content Manager videos 267
 - IT JumpStart intranet 42
 - look and feel 4
 - retrieving complete page from 144
 - retrieving partial page from 144
- Web site designer 49
- Web sites
 - create and update 9
- WebSphere Application Server
 - problem starting 66
 - verify installation 72
- WebSphere log files 64
- WebSphere Member Manager 201
- WebSphere Portal Server 188
- WebSphere portal server 201
- wildcard search 21
- Windows Client 22, 30
- Windows Client tracing log file 64
- WMM 201
- word search 21
- work node 26
- workflow 4, 10–11, 26
 - create 9
 - creating 48
- workflow stage 11
- WPS
 - integrating LWWCM 190

X

- XML 21
- XMLContent
 - viewing data 113



Lotus Workplace Web Content Management and Content Manager Working Together

(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



Lotus Workplace Web Content Management and Content Manager Working Together for LWWCM Java Edition V2 and CM V8 Only

**Using CM as a
repository for
LWWCM**

**Publishing CM
content to LWWCM**

**Covering search,
security,
VideoCharger
integration, and
migration**

In this IBM Redbook, we explore how we can put both IBM Lotus Workplace Web Content Management (LWWCM) V2.0 Java Edition and IBM DB2 Content Manager (Content Manager) V8 products at work together.

We provide overviews of LWWCM and Content Manager products, discuss why we might want to integrate both products and when to do so.

Before you begin integration, we cover various areas that you need to be familiar with and aware of. We provide verification checklists, hints and tips on system installation and configuration.

To create an integrated LWWCM with Content Manager solution, we show you how you can use Content Manager as the data repository for LWWCM and how to publish existing Content Manager data to LWWCM using federated content components, connect tags, and JSP components.

With the basics of how to integrate, we address some of the issues you need to consider when planning the system architecture for the integrated solution.

Other topics included in this redbook are portal integration, search, security, VideoCharger integration, syndication, data migration, and versioning.

This IBM Redbook is ideal for IT architects and specialists who will be working on the integration of LWWCM and Content Manager solution projects.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks