Lotus software

IBM

IBM Workplace Web Content Management

for Portal 5.1 and IBM Workplace Web Content Management 2.5

- Installation, configuration, and deployment best practices
- Building and managing a site
- Advanced topics

Marilyn Flax
Michael Fromin
Amier Jordan
Sabine Nagl
Thomas Radigewski
Theresa Smit
Oliver Trabert

Redbooks

ibm.com/redbooks

Archived

Archived

International Technical Support Organization

IBM Workplace Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5

January 2006

Archived

SG24-6792-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xv.

Archived

First Edition (January 2006)

This edition applies to IBM Workplace Web Content Management, Release 5.1 (included with WebSphere Portal 5.1) and IBM Workplace Web Content Management 2.5.

If you are a customer using IBM Workplace Collaboration Services 2.5, IBM Workplace Web Content Management 2.5 is included. For specific details about the versions and the positioning, refer to 1.3.2, "Understanding the versions and the positioning" on page 11.

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

Contents

Notices xv
Trademarks xvi

Preface xvii
The team that wrote this redbook. xvii
Additional contributors to this book xx
Become a published author xxi
Comments welcome. xxii

Part 1. Introducing IBM Workplace Web Content Management 1

Chapter 1. Introduction to IBM Workplace Web Content Management . . . 3
1.1 Importance of content management 4
1.1.1 Information within the context of Web content 4
1.1.2 What is content? 6
1.2 Overview of Web content management 6
1.2.1 The focus for today's businesses 7
1.2.2 General concepts in Web content management. 8
1.3 Introducing IBM Workplace Web Content Management 9
1.3.1 Introduction to IBM Workplace Web Content Management 10
1.3.2 Understanding the versions and the positioning 11
1.3.3 Benefits of using IBM Workplace Web Content Management 12
1.3.4 The importance of data modeling 14
1.3.5 Highlights of new features since release 2.0. 14
1.4 Setting proper expectations about working with this product 16
1.4.1 The roles involved and their focus 17
1.5 How people use IBM Workplace Web Content Management 21
1.6 Licensing 22
1.7 Overview of chapters in this book 26

Part 2. Deployment planning and installation 31

Chapter 2. Product architecture and deployment planning 33
2.1 IBM Workplace Web Content Management 5.1 Component Architecture 34
2.1.1 Key components 35
2.1.2 Changes from earlier versions 41
2.2 Rendering IBM Workplace Web Content Management Content. 43
2.2.1 An overview of the rendering options 43
2.2.2 Alternate Rendering Concepts 46

© Copyright IBM Corp. 2006. All rights reserved.

2.3 IBM Workplace Web Content Management environment types 47
2.3.1 Development 47
2.3.2 Authoring 48
2.3.3 Staging 48
2.3.4 Production (live) 49
2.3.5 Workflow versus syndication 49
2.3.6 Putting the environments together 52
2.4 Sample physical architectures 53
2.4.1 Basic 54
2.4.2 Intermediate 55
2.4.3 Advanced architecture planning 57
2.5 Example infrastructure 63
2.5.1 Environments 63
2.5.2 The River Bend infrastructure 65

Chapter 3. Installing and configuring IBM Workplace Web Content Management 69
3.1 Installation overview 70
3.1.1 The IBM Workplace Web Content Management Server components for this book 71
3.1.2 Major Installation Changes from IBM Workplace Web Content Management 1.0 and later and 2.0 and later 71
3.2 Installing IBM Workplace Web Content Management 73
3.2.1 Base server installation 73
3.3 Configuring IBM Workplace Web Content Management with DB2 Universal Database 77
3.3.1 Creating an IBM Workplace Web Content Management database on DB2 server 78
3.3.2 Configuring DB2 Connect Client on WebSphere Portal and IBM Workplace Web Content Management server 79
3.3.3 Changing the configuration 82
3.3.4 Confirming your new IBM Workplace Web Content Management configuration 86
3.4 Applying fixes for IBM Workplace Web Content Management 5.1 87
3.4.1 Technote, reference #1201178 87
3.4.2 IBM Workplace Web Content Management connect.cfg file 91
3.4.3 IBM Workplace Web Content Management fix PK02213 92
3.5 Installing the Rendering Portlets 95
3.5.1 Local Rendering Portlet 95
3.5.2 Remote Rendering Portlet 107
3.6 Post-installation configuration 112
3.6.1 E-mail 112
3.6.2 Logging levels 113

Archived

3.6.3 Caching . . . 114
3.7 Static pre-rendering . . . 114
3.7.1 Enabling the correct caching subsystem . . . 115
3.7.2 Configuring the <Cacher> configuration options . . . 117
3.7.3 Manually running the pre-rendering engine . . . 119
3.7.4 Known issues with pre-rendering in IBM Workplace Web Content Management 5.1 . . . 119
3.8 Removing IBM Workplace Web Content Management . . . 122

Chapter 4. Overview of IBM Workplace Web Content Management 2.5 . 125
4.1 Introduction to IBM Workplace Web Content Management 2.5 . . . 126
4.2 Functional differences between IBM Workplace Web Content Management 2.5 and 5.1 . . . 126
4.2.1 Supported browsers . . . 126
4.2.2 Supported Java Virtual Machines . . . 127
4.2.3 Portal Document Manager integration . . . 127
4.2.4 ODC (Rich Text) editors . . . 127
4.2.5 Fix packs and iFixes . . . 127
4.2.6 Portal Personalization (available with 5.1.0.1 and later) . . . 128
4.3 Installing IBM Workplace Web Content Management 2.5 . . . 128
4.3.1 Installing IBM Workplace Web Content Management 2.5 on Windows 2000 Advanced Server . . . 129
4.4 Configuring IBM Workplace Web Content Management 2.5 . . . 137

Part 3. Key concepts, information architecture, site design, and reference implementations for Web content management . . . 139

Chapter 5. Key concepts and terminology . . . 141
5.1 Separating content and presentation . . . 143
5.1.1 Content editing considerations . . . 146
5.2 Componentization and the page design concept . . . 149
5.2.1 Considerations for reusability . . . 151
5.3 Key IBM Workplace Web Content Management functions, items, and definitions . . . 152
5.3.1 Version management . . . 155
5.3.2 Syndication . . . 157
5.3.3 Item and component security . . . 158
5.3.4 Authoring template . . . 161
5.3.5 Category Management . . . 164
5.3.6 Workflow . . . 166
5.3.7 Content . . . 169
5.3.8 Site framework . . . 170
5.3.9 Design and development components . . . 174
5.3.10 Search rules . . . 194

5.4 Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1 . . . 196

Chapter 6. Information architecture and site design . . . 199
6.1 Defining the information architecture . . . 200
6.1.1 Base example of a good information architecture . . . 201
6.1.2 Results of an inadequate information architecture . . . 215
6.2 Key considerations and decision processes . . . 216
6.2.1 Understand the goals . . . 217
6.2.2 Understand your audience . . . 217
6.2.3 Consider objectives and key success factors . . . 218
6.2.4 Find the killer app . . . 219
6.2.5 Consider a user-centric design . . . 219
6.2.6 Define use case scenarios . . . 222
6.2.7 Plan the site framework and site areas . . . 223
6.2.8 Metadata, keywords, and categories . . . 228
6.2.9 Determining the components of the Web site . . . 229
6.2.10 Using workflow and syndication . . . 230
6.2.11 Using syndication . . . 231
6.3 Information design . . . 232
6.3.1 Using presentation templates . . . 233
6.3.2 Using site navigators . . . 234
6.3.3 Using menus . . . 238
6.3.4 Using direct feature links . . . 241
6.3.5 Using embedded hyperlinks . . . 243
6.3.6 Using search . . . 244
6.4 Site acceptance . . . 249
6.4.1 Business stakeholders . . . 249
6.4.2 Design and layout approach . . . 249
6.4.3 Communications and enablement . . . 250
6.4.4 Wireframe demonstration . . . 250
6.5 Workshop approach . . . 250
6.5.1 Workshop preparation . . . 250
6.5.2 The workshop . . . 251
6.5.3 Work products . . . 251

Chapter 7. Building the River Bend Web site with IBM Workplace Web Content Management . . . 253
7.1 An overview of the scenario . . . 255
7.2 Overview of the process . . . 256
7.3 Prerequisites for building the example Web site . . . 257
7.4 Accessing IBM Workplace Web Content Management . . . 257
7.5 Building the site framework . . . 259

7.5.1 Creating the site 259
7.5.2 Creating the site areas 260
7.5.3 Rearranging the site area order 265
7.6 Building the taxonomy and categories 266
7.6.1 Creating the site taxonomy 266
7.6.2 Creating categories for the site taxonomy. 267
7.7 Building a workflow 268
7.7.1 Creating the intranet workflow stages 268
7.7.2 Creating the publish stage 269
7.7.3 Creating the Development Express Workflow. 270
7.7.4 Creating the Approval Workflow 271
7.7.5 Creating actions 272
7.8 Creating an authoring template 274
7.8.1 Creating the River Bend Development authoring template 274
7.8.2 Copying the River Bend Development authoring template 276
7.9 Creating the presentation templates 276
7.9.1 Creating the presentation templates 277
7.9.2 Creating the two-column table presentation template 279
7.9.3 Creating the two-column table search presentation template 281
7.10 Pairing the authoring template and the presentation templates 283
7.11 Creating and referencing image components 285
7.11.1 Creating image library components 285
7.11.2 Referencing images in a presentation template 287
7.12 Creating content 288
7.12.1 Creating content to test that items in the site work properly 289
7.12.2 Creating a content item and assigning it as the default content page for a site area 291
7.12.3 Referencing an image component in a content item. 292
7.12.4 Creating content items for the remaining site areas in the River Bend Web site 293
7.12.5 Opening and viewing the River Bend Web site in a browser 303
7.13 Using stylesheets 303
7.13.1 Saving the stylesheet as a file resource component. 304
7.13.2 Referencing the stylesheet in the presentation templates 304
7.14 Adding a left navigator component 305
7.14.1 Creating a left navigator component 305
7.14.2 Referencing the left navigator in the presentation templates 307
7.15 Formatting the navigator 308
7.15.1 Creating HTML components to specify formatting 308
7.15.2 Referencing the HTML formatting components in the left navigator component 309
7.16 Creating a breadcrumb navigator 310
7.16.1 Creating the breadcrumb navigator component 310

7.16.2 Referencing the breadcrumb navigator component in the presentation templates . . . 311
7.17 Creating a site map navigator . . . 312
7.17.1 Creating the site map navigator component . . . 312
7.17.2 Displaying the site map navigator as component in the Site Index content item . . . 313
7.18 Creating menus . . . 314
7.19 Creating a site area menu component . . . 314
7.19.1 Referencing site area menu in specified content pages . . . 316
7.19.2 Creating an HTML component to format the site area menu . . . 317
7.19.3 Referencing the menu format in the site area menu's component design . . . 317
7.20 Creating a menu using categories . . . 319
7.20.1 Referencing the category menu in the Espresso site area . . . 319
7.20.2 Referencing the category menu in the Three Column Table presentation template . . . 320
7.21 Creating the related information menu . . . 321
7.21.1 Referencing the related information menu in the Three Column Table presentation template . . . 322
7.22 Creating an edit document navigator . . . 323
7.22.1 Referencing the edit document navigator component in the presentation templates . . . 324
7.23 Creating a tools navigator . . . 325
7.23.1 Referencing the tools navigator component in the presentation templates . . . 326
7.24 Enabling the Search module . . . 327
7.24.1 Editing the connect.cfg file . . . 327
7.24.2 Editing the aptrixsearch.properties file . . . 328
7.24.3 Modifying a presentation template for the Search page . . . 329
7.24.4 Creating a search results template . . . 330
7.24.5 Creating an HTML search form . . . 331
7.24.6 Referencing the HTML search form in the Search content item . . . 332
7.24.7 Creating an HTML quick search form . . . 333
7.24.8 Referencing the HTML quick search form . . . 334
7.25 Viewing content in a Portlet . . . 334
7.25.1 Creating a Portal Page Layout presentation template . . . 334
7.25.2 Creating and copying Content Viewer Portlets . . . 336
7.25.3 Creating a Portal Page and adding the Content Viewer Portlets . . . 337
7.25.4 Configuring Portlets . . . 338
7.26 Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1 . . . 341
7.26.1 Configuring and creating library, folder, and documents in Portal Document Manager . . . 341

7.26.2 Creating an authoring template with a Portal Document Manager component . . . 343
7.26.3 Referencing the new component in the presentation template . . . 344
7.26.4 Pairing the Portal Document Manager authoring template and the presentation template . . . 344
7.26.5 Creating content using the River Bend Portal Document Manager authoring template . . . 345

Chapter 8. Caching and pre-rendering considerations . . . 347
8.1 Improving performance . . . 348
8.1.1 Caching options . . . 349
8.1.2 Pre-rendering options . . . 351
8.2 Implementing a caching or pre-rendering strategy . . . 353
8.3 Setting up caching . . . 356
8.3.1 Understanding expiration parameters . . . 356
8.3.2 Configuring the default cache . . . 357
8.3.3 Understanding cache size . . . 358
8.3.4 Configuring basic caching . . . 359
8.3.5 Configuring advanced cache . . . 359
8.4 Configuring data caching . . . 362
8.5 Configuring pre-rendering . . . 363
8.5.1 Pre-rendering via an IBM Workplace Web Content Management local rendering server . . . 364
8.5.2 Pre-rendering via an HTTP Server . . . 365
8.6 Custom caching with connect tags . . . 368
8.7 Considerations in developing a caching strategy . . . 371
8.7.1 Identifying performance impacts . . . 371
8.7.2 Cache recommendations for components types . . . 373
8.8 Other caching systems that can effect performance . . . 375
8.9 Documenting the cache strategy . . . 376
8.9.1 Component and template catalog . . . 376
8.10 WebSphere Dynamic Cache Service . . . 380
8.10.1 High performance (caching) options . . . 380
8.10.2 Dynamic cache install and configuration . . . 385

Chapter 9. Advanced integration with WebSphere Portal . . . 403
9.1 Overview of the WebSphere Portal integration . . . 404
9.2 Reasons for using Portal navigation . . . 405
9.2.1 Understanding the layout of the Portal page . . . 407
9.3 Bridging the gap . . . 408
9.3.1 Explicit mapping . . . 408
9.3.2 **Implicit mapping . . . 410**
9.3.3 Choosing structures that make mapping easy . . . 410

Archived

9.4 Beginning the implementation 414
9.4.1 Explicit mapping approach 414
9.4.2 Making the implementation more functional 416
9.5 Implementing a custom display Portlet 418
9.6 Managing links 418
9.6.1 Understanding links inside of the Portal framework 419
9.6.2 Implementing the links using 2-Phase links 421
9.6.3 Implementing the link management 422
9.6.4 Building the links 423
9.6.5 Walking through of the linking process 424
9.7 Using multilingual and personalized content display 425
9.7.1 Ensuring link integrity 427
9.7.2 Other linking functionalities 427

Chapter 10. Search functionality for IBM Workplace Web Content Management 429
10.1 Providing multiple techniques to search content 430
10.2 Using the IBM Workplace Web Content Management search module . 430
10.2.1 Security and the search module 431
10.2.2 The search module index 431
10.3 Using the IBM Workplace Web Content Management API 432
10.4 Using the WebSphere Portal search engine 433
10.4.1 Crawling a WebSphere Portal site 433
10.4.2 Considerations for using the WebSphere Portal search engine . . 435
10.4.3 Crawling a regular Web site 436
10.4.4 Crawling using a custom method 436
10.4.5 Using multiple Portal pages to display content 439
10.4.6 Considering outside search engines 439
10.5 Using external and third-party search products 440
10.5.1 Key considerations when select a third-party search product 441

Chapter 11. Personalization rules for IBM Workplace Web Content Management 443
11.1 Overview of the personalization feature 444
11.2 Installing and configuring WebSphere Portal 5.1.01 446
11.3 Enabling WebSphere Portal personalization 446
11.3.1 Changing the connect.cfg file 446
11.3.2 Using personalization rules on IBM Workplace Web Content Management Content 447
11.3.3 Populating the JCR with existing content information 448
11.3.4 Creating a personalization rule 449
11.3.5 Configuring the Personalized List portal 451
11.3.6 Including rules on presentation template 453

11.3.7 Creating rules 457
11.3.8 Creating IBM Workplace Web Content Management content using a profile 461
11.3.9 Configure the Portlet in personalization demo 464
11.3.10 Including the rule in JSP 467

Chapter 12. Working with the application programming interface 469
12.1 Introduction to the API 470
12.1.1 Use cases for the API 470
12.2 Components of the IBM Workplace Web Content Management API . . 471
12.2.1 The WCM_API class 474
12.2.2 The repository interface 474
12.2.3 Workspace interface 475
12.2.4 Document object 476
12.2.5 DocumentID interface 477
12.2.6 DocumentType and DocumentTypes interfaces 477
12.2.7 RenderingContext interface 478
12.2.8 Editable components of the interfaces 478
12.2.9 Read-only component interfaces 482
12.2.10 UserProfile 485
12.2.11 Iterator interfaces 485
12.3 Extending your Web site with JSP support 485
12.3.1 JSP components 486
12.3.2 JSP tag library 487
12.3.3 JSP tag library example 490
12.4 Setting up the development environment 491
12.4.1 Installing Rational Application Developer 6.0 492
12.4.2 Installing the API Java files 496
12.4.3 Attaching to a remote server 497
12.5 Examples of API development 501
12.5.1 Developing a simple Portlet 501
12.5.2 Working with JSP components 526
12.6 Deployment considerations 530

Chapter 13. Advanced topics and examples with the API 533
13.1 Multi-lingual Content Display Portlet 534
13.2 Using the Customizable Template Portlet 535
13.2.1 Installing the portlets 538
13.2.2 Creating a Bulletin Board page with portlets 540
13.2.3 Configure the Portlets 542
13.2.4 Demonstration of Quick Edit 548
13.2.5 Customizing the Content Custom UI Portlet 550
13.2.6 Customizing the Workplace Web Content Management Sample

Workflow Portlet 561
13.3 Extensions to the Customizable Template Portlet. 562

Chapter 14. Migrating to IBM Workplace Web Content Management 5.1 565
14.1 Overview of migrating data from a previous version 566
14.1.1 Differentiating upgrading, transferring, and migrating 567
14.1.2 Upgrading IBM Workplace Web Content Management 567
14.2 Transferring file system data 568
14.2.1 Transferring file system data 569
14.2.2 Using the wpconfig.properties file during data transfer 570
14.2.3 Running the data transfer task 570
14.2.4 Ensuring valid data format 570
14.3 Verifying and troubleshooting file system data transfer. 571
14.3.1 Verifying the data transfer 571
14.3.2 Examining the error messages 571
14.3.3 Known limitations 573
14.3.4 Files that are needed by the IBM Support Desk 573
14.4 Migrating users and groups. 573
14.4.1 Migrating users 574
14.4.2 Creating migrated users and groups with WebSphere Member Manager 576
14.4.3 Migrating users and groups. 576
14.4.4 Migrating to WebSphere Member Manager using a user migration mapping file 577
14.4.5 Adding new users and groups to WebSphere Member Manager using the migrationUsers.xml file 578
14.4.6 Migrating the Administrators group 578
14.4.7 Migrating LDAP users and groups 579
14.4.8 UI Access Group 579
14.4.9 ALL USERS group 579
14.4.10 Anonymous user 580
14.4.11 Member Fixer utility. 580
14.5 Verifying and troubleshooting user and group migration. 581
14.5.1 Verifying user migration 581
14.5.2 Using the log files 581
14.5.3 Examining common errors and resolutions. 582
14.5.4 Files needed by the IBM Support Desk. 584
14.6 Creating the unrestricted resources 584
14.6.1 Where resources are stored 585
14.6.2 Accessing unrestricted resources 585
14.6.3 Resource Checker utility 586
14.7 Migrating data from a previous version 587

14.7.1 Transferring file system data to a Cloudscape database 587
14.8 Enabling user migration. 590
14.9 Migrating users directly to WebSphere Member Manager 591
14.10 Migrating users to WebSphere Member Manager using a user migration mapping file 593
14.10.1 Generating a user migration mapping file 593
14.10.2 Reviewing and editing the user migration mapping file. 594
14.10.3 Processing the user migration mapping file 595
14.10.4 Verifying that the migration was successful 595
14.10.5 Adding new users or groups to WebSphere Member Manager . 596
14.11 Disabling user migration 597
14.11.1 Recreating the resources 597

Chapter 15. Migrating content into IBM Workplace Web Content Management 599
15.1 A roadmap to migrating Web content 600
15.1.1 Determining the business case for the migration 601
15.1.2 Determining the migration approach and toolset 602
15.1.3 Determining source classifications for the migration. 603
15.1.4 Determining the target definition for the migration 604
15.1.5 Determining the transformation requirements for the migration . . 605
15.1.6 Determining quality assurance measures 607
15.1.7 Undertaking the migration 608
15.2 Overview of the Vamosa Content Migrator 609
15.2.1 How Vamosa Content Migrator works. 609
15.2.2 Reasons for using a specialized method 614
15.2.3 The law of diminishing returns 616
15.2.4 Quality assurance and project management. 616

Appendix A. Additional material 617
Locating the Web material 617
Using the Web material 618
How to use the Web material 619

Related publications 621
IBM Redbooks 621
Online resources 621
How to get IBM Redbooks 622
Help from IBM 622

Index 623

Archived

Archived

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.

© Copyright IBM Corp. 2006. All rights reserved.

Trademarks

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

- @server®
- @server®
- Redbooks (logo) ™
- developerWorks®
- i5/OS®
- z/OS®
- Ascendant®
- Cloudscape™
- Domino®
- DB2 Connect™
- DB2 Universal Database™
- DB2®
- IBM®
- Lotus Notes®
- Lotus Workflow™
- Lotus®
- Notes®
- OmniFind™
- Rational®
- Redbooks™
- WebSphere®
- Workplace™
- Workplace Web Content Management™

The following terms are trademarks of other companies:

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

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

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

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

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

Archived

Preface

IBM® Workplace™ Web Content Management is a powerful and sophisticated Web content management product that is designed to accelerate development and delivery of critical business information. It enables end-to-end collaboration for content creation, approvals, management, retention, and publishing across Internet, intranet, extranet and portal assets. Despite its power, it is remarkably easy to use. It simply requires standard office skills to harness and optimize the flow of information rapidly.

This IBM Redbook provides a comprehensive approach to better understanding and using Workplace Web Content Management™. It addresses the specific technical aspects of IBM Workplace Web Content Management, by discussing its underlying architecture and features, by recommending deployment scenarios, and by providing detailed discussions on how to best use the product and to take advantage of built in features and its extensible API.

To address the needs of a broad range of users, from those new to the product to those with much more experience, we first provide a basic tutorial on building a site. We then build upon this foundation to explore advanced topics and opportunities for customization. In addition to discussing specific technical features, this book also discusses the key concepts and best practices for creating a accurate Information Architecture and site framework. Finally, this book addresses how to upgrade IBM Workplace Web Content Management data from earlier versions of the product, as well as how to migrate data in from external Web content management systems.

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 (ITSO), Cambridge, Massachusetts Center.

John Bergland is a project leader at the International Technical Support Organization, Cambridge Center. He manages projects that produce Redbooks™ about Lotus® Software products. Before joining the ITSO in 2003, John worked as an Advisory IT Specialist with IBM Software Services for Lotus (ISSL), specializing in Notes and Domino® messaging and collaborative solutions.

© Copyright IBM Corp. 2006. All rights reserved.

Marilyn Flax is a Learning Specialist in Sydney, Australia. She has 10 years of experience in user interface design and usability, and course development and technical writing. She has worked with IBM Workplace Web Content Management for the past six years and has worked at IBM for two years. She has written extensively on IBM Workplace Web Content Management, including user guides, instructor-led training materials, and e-learning and has used the materials to deliver face-to-face and remote Internet-based training to customers, business partners, and IBM employees.

Michael Fromin is Director of Education and Training for Web Content and WebSphere® Portal Practice Manager Ascendant® Technology (`http://www.atech.com`). Michael has been in technology for over 15 years, 10 of which were in IT management for two Fortune 50 companies. Michael has held lead management responsibility for Network Architecture, Help Desk Management, and Development. For the past eight years, he has focused on leveraging the IBM and Lotus software portfolio to help companies improve business systems and processes. Prior to Ascendant, Michael managed the services team for CONET USA, where he and his team focused on delivering Web content management solutions based on Lotus Domino, Java™, and the you@web product line. In addition to his real-world experience, Michael has authored numerous technical articles that focused on advanced development techniques for Lotus Domino, Java, and WebSphere Portal.

Amier Jordan is a WebSphere Portal and IBM Workplace Web Content Management expert who has worked as an IT consultant and business consultant since 1998. He has worked with over 50 different customers in various industries with accounts. His recent work was involved with the Portal project which won the Lotus Award for *Best Portal Solution 2005*. His key skills are in On Demand Workplaces, Portals, IBM Workplace Web Content Management, and other Web Content Management technologies, and finally, Enterprise application integration. Other areas of his expertise include project management and Lotus Notes® development. Amier holds a Master of Engineering degree from the Technical University of Darmstadt, Germany.

Abstract

Sabine Nagl is an IT Specialist at the IBM Software Innovation Team (SWIT) for Lotus Products in Germany. She has more than 12 years of experience in the IT industry and holds a masters degree in Computer Science from the Technical University in Munich. Sabine joined Lotus in 1998 as a Presales IT Specialist working with Integrated Accounts as well as SMB customers, primarily in Germany but also in Europe. She started as technical expert for Domino and Domino Family products and soon specialized in Document Management, Knowledge Management, and Workflow Management. Sabine is frequently a technical speaker at IBM and IBM Partner events.

Thomas Radigewski is the founder of RADIGEWSKI Informatik in Germany (`http://www.radi.info`). Thomas holds a degree in Business Information Technology from the University of Applied Science Cologne, Gummersbach Campus. He started his professional career with CONET AG system integrators and IT consultancy. His responsibilities included software development, project management, and the overall direction of the development of CONET's own you@web software brand based on Lotus Notes and Domino. Thomas has extensive project and development expertise in content management, integration, and groupware projects.

Theresa Smit has a wide breath of application development knowledge in her more than 28 years in computing technologies. Currently, she is part of the IBM Portal development organization, working with external and internal customers to use portal technologies to manage and deliver Web content. In addition to the Portal product, she also supports the use of IBM Lotus Workplace Web Content Management, WebSphere Portal Content Publishing, and Personalization products. She has authored other internal and external articles and has presented at developerWorks®, Portal, Lotus, and IBM Content Manager technical conferences.

Oliver Trabert is an IT Architect for WP Experts in Germany. He has worked exclusively with WebSphere Portal for more than three years. His recent Portal project won the Lotus Award for *Best Portal Solution 2005*. One of his specialties is the WebSphere Content Management and Portal integration. Oliver is also co-author of multiple other IBM Redbooks on WebSphere Portal and is a frequent speaker at IBM events.

Additional contributors to this book

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

Kamie Ngam-Nield has over 12 years experience in the IT industry. She joined IBM United Kingdom in 2000. Kamie is responsible for technical leadership on a number of key Domino and Web application deployments within the financial and public sectors. A co-author of the first LWWCM Redbook, Kamie also authored a number of presentations, articles on WebSphere Content Management architecture and implementation best practices.

Nic Archer is a Product Champion for Vamosa. Nic has been involved in the technology sector since 1985. He joined Computer Associates in 1987 as a systems engineer and has worked in many areas of system automation and management, such as workload scheduling, performance monitoring, security and audit. He became a product manager for security products in 1993. In 1994, he co-founded Gamut Technologies. Nic worked as a Siebel Partner manager at Druid until their acquisition by Xansa in 2000, and he joined Vamosa in 2002. Nic is responsible for the go-to-market strategy and the product management of the Vamosa product range and also for the Vamosa brand.

Stefan Freudenberger is a Portal developer for WebSphere Portal Experts in Germany. He has worked exclusively with WebSphere Portal for more than two years. He also worked on the Portal project that won the Lotus Award for *Best Portal Solution 2005.*

Mark Osborn, EMEA Sales Leader for Lotus Content Management

A special thanks to Mark Osborn, as he provided clarification around product strategy, positioning, and licensing issues.

Jason Hatch, IBM Sales and Distribution

Howard Berman, IBM Software Group, Lotus Software Engineer

Herbert Hilhorst, IBM Software Group, IBM Certified IT Specialist. Herbert is based in France and has been working with IBM Lotus since 1994, covering support, services, and technical sales. He has a Master of Science in Electrical Engineering from Twente University, The Netherlands. Currently, he is working for the pan-EMEA Technical Sales team where he drives Workplace, Portal, and Collaboration Software.

Archived

Evan Borysko, Senior Consultant, Ascendant Technology (http://www.atech.com)

Robin King, Senior Consultant, Ascendant Technology (http://www.atech.com)

A special thanks to Evan Borysko and Robin King, as they contributed much of the caching content for Chapter 8, “Caching and pre-rendering considerations” on page 347.

Chris Knight, IBM Software Group, Lotus Software Engineer

Doug Merhar, IBM Software Group, Lotus, IBM Lotus Workplace, and Web Content Management Level 2 Support

A special thanks to Chris Knight and Doug Merhar, as they answered many questions on patches and other features of IBM Workplace Web Content Management.

Tony Higham, IBM Software Group, Lotus Portal, and Content Management Specialist, IBM, Atlanta, Georgia

Steve Garward, IBM Software Group

Jen Wang, IBM Web Content Management Product Development and Support, IBM, Raleigh, North Carolina

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 will team with IBM technical professionals, Business Partners, and customers.

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

Archived

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 e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, New York 12601-5400

Archived

Part 1

Introducing IBM Workplace Web Content Management

Archived

© Copyright IBM Corp. 2006. All rights reserved.

Archived

1

Introduction to IBM Workplace Web Content Management

IBM Workplace Web Content Management is a powerful and sophisticated Web content management product that is designed to accelerate development and delivery of critical business information. It enables end-to-end collaboration for content creation, approvals, management, retention, and publishing across Internet, intranet, extranet and portal assets. Despite its power, it is remarkably easy to use.

This book addresses the specific technical aspects of the IBM Workplace Web Content Management product by discussing its underlying architecture and features, by recommending deployment scenarios, and by providing detailed discussions on how to best use the product. To address the needs of a broad range of users, this book first provides a basic tutorial on building a site. It then builds upon this foundation to explore advanced topics and opportunities for customization. In addition to discussing specific technical features, this book discusses the key concepts and best practices for creating a accurate information architecture and site framework. Finally, this book addresses how to upgrade IBM Workplace Web Content Management data from earlier versions of the product, as well as migrate data from external Web content management systems.

© Copyright IBM Corp. 2006. All rights reserved.

1.1 Importance of content management

Prior to discussing specific aspects of the importance of *Web content management,* we begin by discussing the underlying importance of content management. The ability to effectively manage content — and more importantly information and specific knowledge — directly affects your organization's ability to succeed in the marketplace. Business information is, in most cases, the most valuable asset that you offer to your customers, employees, or business partners. Accordingly, that makes information, together with the ability to manage it, integral to the success of every business.

Value is the knowledge that you add *before* you offer something. So, business value is based on the information that you share. This concept is true both for the production as well as for knowledge and service industries. Looking at where this information exists, you find that it is often divided and decentralized throughout organizations. *Information is stored frequently as content.* For example, critical information can include product descriptions, instructions, policies, key customer issues, competitive strategies, and so forth. The effectiveness of the information you offer — and in many cases the value you provide — depends directly upon the ability to distribute that information in a timely and efficient manner.

1.1.1 Information within the context of Web content

When your organization offers information via a Web browser, this material is considered *Web content*. When thinking about Web content, it is important to note that Web content consists of two parts:

- The content
- The design or presentation of that content

Going forward, we explain this concept in greater detail and discuss the advantages of handling the content and the design of the content separately.

There are two approaches to creating, managing, and publishing Web content:

- Static Web content

  Static Web content is content that is embedded directly into the design and placed statically on a Web page. This type of content is most likely the result of programming rather than content authoring. Due to its static nature and embedded design, this type of content can be difficult to reuse and change.

  Static Web content solutions often require someone within an IT department, or at least a person who possesses Web design and some Web-based IT skills, to translate content into HTML before it can be posted to a Web site or a portal. Accordingly, static Web content is more difficult to use within a dynamic Web site — a Web site that changes frequently and that requires

up-to-date information to deliver maximum value. Additionally, static Web content is often impossible to reuse due to its embedded design.

- Dynamic Web content

  Rather than embedding the information directly together with the design, *dynamic Web content* treats the content and the presentation of the content as two distinct entities. The separation of Web content from the presentation layer allows you to manage and deliver content quickly and efficiently, independent of its presentation. The ability to manage both content and its presentation layer as separate entities is enabled most frequently via Web content management systems.

IBM Workplace Web Content Management allows companies or organizations to leverage the flow of information. By getting information into the hands of the right people easily and efficiently, it is a key component in sustaining a competitive advantage. Authors do not face the technical side of Web technologies. Content that is created information can be integrated with multiple uses and processes — distributed or shared throughout and beyond an organization. For example, a single product sheet can contain complete pricing information, including internal prices which you only wish to share *within* your organization. By taking advantage of the Multi publishing feature, this allows you to publish a subset of exactly the same document for your customers, *without the need to maintain to separate documents*.

Content Management within the context of applications

Finally, it is also important to consider a different view on Content Management that is often not obvious. Applications offered through a Web browser also leverage content that is managed by a content management system. For example a product catalogue or online store contains content that is accessed through an application which (usually) provides a powerful and helpful navigation tool.

However, with ever-growing volumes of critical information distributed across Internet portals, e-business applications, intranets and extranets in various formats, deploying and maintaining effective content management methodologies can be a significant challenge.

Web content management is not just about customers. It is important to employees, business partners, suppliers, and vendors because these groups represent the backbone of an organization. Making sure that they are equipped with the right information and can do business with you more easily than they can with your competitor down the street is critical. A positive Web experience motivates the user to revisit the site like nothing else. This is a challenge that is generally underestimated by most companies.

To be able to compete in today's markets, on demand availability of information is crucial. Content management is the key. Such a powerful management tool, reaching a wide spread audience, also exposes the risks associated with providing information to a broad audience. This makes it necessary to include workflow, security and structure to reflect organization policies in a Web environment.

1.1.2 What is content?

Content can mean many different things, so we should start from a common understanding of what the scope of business content is.

Content supports the work the enterprise does and interacts in many ways with business operations. Enterprise Content Management is the discipline involved with the capture, storage, and management of this kind of content across the enterprise. It makes information easy to find, use, update, and discard when the time comes.

Content can be categorized into a number of types, such as:

- High-volume production content: document imaging and computer output, archiving, and presentation
- Rich media: audio, video, and photos
- Web content: Internet, intranet, and extranet
- Collaborative content: office documents, discussions, and e-mail

The objectives of enterprise content management are:

- To provide an efficient and secure solution for managing content within the enterprise, including enterprise-wide content storage, access, search, distribution, and retention (this is especially true with the current focus on corporate accountability and regulatory compliance).
- To build knowledge-based environments to leverage corporate know-how and expertise.

1.2 Overview of Web content management

In the past, different forms of information — such as Web content, e-mail, product information or customer data — were kept in separate, disconnected content management repositories. Today, that is not a viable solution. To thrive in an on demand environment, you need flexible, cost-effective content management solutions to manage any type of information, including structured data and unstructured content. You need content management solutions that

enable data and content to be integrated with multiple applications and processes - distributed or shared throughout and beyond your organization. Furthermore, you need it structured and organized in the way that is best suited for your business.

1.2.1 The focus for today's businesses

To manage the exponential growth of business-critical information, respond faster to the marketplace and increase employee productivity, your content management capabilities must address three key areas:

- Responsiveness

  Streamlining operations and providing an integrated view of all forms of information can improve your customer relationships and enable your employees, partners and suppliers to work together more efficiently. It can also help you reduce costs and capitalize on new opportunities.

- Productivity

  When employees spend more time looking for information they need to perform their jobs - documents, e-mail, reports, Web content and more, they spend less time working and generating business; if they are not working with the latest information they can miss opportunities and make costly mistakes. When customers cannot find what they are looking for on a Web site, they leave frustrated and dissatisfied. No matter how good the information or prices might be. That can add up to a lot of wasted hours and missed opportunities. The effort to bring visitors back again is costly. Companies that give their employees a simplified, personalized and easily accessible user interface to access and share critical information enjoy a competitive advantage.

- Compliance

  Effectively manage your content to address the increasing demands of government and industry regulations while staying focused on your business goals and using compliance as a lever for change to affect process improvement and business transformation.

Looking at existing enterprises, we find that numerous specialized applications and databases are often already established when it comes to business-critical information. IBM Workplace Web Content Management becomes increasingly important for companies and organizations to leverage the complete spectrum of data assets within their enterprise and share it electronically among suppliers, customers, partners and employees. Content that is created and managed from authors for the propose of Web publishing becomes only one component. Enterprise Content Management (General ECM) provides a basis for the next generation of data management, enabling customers to collect, manage and

Archived

distribute all forms of business information. An employee of a human resources department updating a phone number in an internal application for example, has actually executed a step within a greater content management context, namely the changed number was simultaneously published to the Web. It is mandatory to integrate content from a broad range of existing data sources. Open standards for content exchange are the key.

1.2.2 General concepts in Web content management

This section defines some concepts that are fundamental to understanding Web content management and more importantly, understanding the benefits which can be gained.

- Separation of content and presentation

  As a foundation for understanding the benefits of IBM Workplace Web Content Management, it is first necessary to distinguish between content and the presentation of the content. When a clear separation is made between content and how it is presented, you can appreciate how a single piece of content could potentially be rendered in multiple ways. For example, a single item of content such as a news article could be presented in any of the following ways:

  - On a Web page in a variety of formats, based on user preferences and personalization
  - In a PDF file
  - Downloaded to a PDA
  - Streamed as an RSS feed

  The content is the same, but the presentation can be adapted to best meet a user's needs within the context of their specific role or preference.

  This approach also guaranties the consistent look and feel of a Web site. If the design changes, the content parts remain unaffected. Additionally, the content creators do not have to worry about the presentation of their content. This is important because content creators usually do not have significant knowledge of HTML.

  Moving forward with this idea, key concepts and functional areas that apply to IBM Workplace Web Content Management can be grouped in terms of dynamic presentation or content management.

- Content creation and authoring

  Creating the content and information that is intended to be delivered. Content creators are guided through the authoring process — using familiar applications — without having to learn new technical skills.

- Workflow

  The accuracy, relevancy, and recency of content is assured by automating the content life cycle — from creation through approval and delivery to review and archival.

- Management

  Content becomes an asset again as control of the site is placed back in the hands of the users who understand the subject and customers best.

- Content publication

  When content has been authored and approved, this is the stage which releases the content for delivery to the live site. This can be a simple process, such as making a file available on a file system to a Web server and advertising the URL, or it can be a more complex procedure such as moving content through a complex workflow and into a production environment.

- Content aggregation

  This occurs when content from various sources is brought together. In a simple Web site this actually occurs manually during the content creation phase. However, in a more dynamic and data-driven environment the aggregation occurs at an application server level.

- Integration

  Aggregation, transactional integration and performance enhancing caching enrich the user experience.

- Content delivery

  Content needs to be delivered to the user in the appropriate (or desired) format. Most frequently, this implies delivery via HTTP using browsers and similar devices. Alternatively, content delivery could refer to other publish and subscribe methods and data feeds or Web service protocols.

1.3 Introducing IBM Workplace Web Content Management

IBM Workplace Web Content Management software comes in two varieties — one for small and medium-sized business — and the other for large and very large organizations:

- IBM Workplace Web Content Management is for enterprise-sized organizations
- IBM Workplace Web Content Management, Standard Edition is designed specially for small to medium-sized businesses

Archived

Note: The product runs on both Lotus Domino and IBM WebSphere. In this book, we will deal with the Java Edition of the product because the Domino Version is been described in detail in *Lotus Workplace Web Content Management*, SG24-6309, which is available at:

http://www.redbooks.ibm.com/abstracts/sg246309.html?Open

There are subtle differences between IBM Workplace Web Content Management, Version 5.1 and IBM Workplace Web Content Management, Version 2.5 editions. In this book we focus primarily on IBM Workplace Web Content Management, Version 5.1 but also address the differences in IBM Workplace Web Content Management, Version 2.5 as applicable. (See 1.3.2, “Understanding the versions and the positioning” on page 11 for more details on the versions and the positioning of each.)

1.3.1 Introduction to IBM Workplace Web Content Management

IBM Workplace Web Content Management is a key component of the IBM Workplace software solution, enabling users at all levels to manage end-to-end Web content creation and information life cycles, personalization and publishing for dynamic intranet, extranet, Internet and portal sites. The IBM Workplace software solution strategy, an IBM strategy that integrates vital portal, collaboration, and Web content management infrastructure, enables the creation of an on demand environment for human interaction. IBM Workplace software product functionality includes team collaboration, messaging, document management, e-learning and a wide range of business process and applications.

By leveraging content in backend systems, IBM Workplace Web Content Management reduces development and implementation time and places content creation and management firmly in the hands of content experts for *author once, publish everywhere* control. The product runs on both Lotus Domino and IBM WebSphere platform and provides for the integration of IBM WebSphere Portal and IBM DB2® Content Manager - making this offering a great IBM middleware fit and an opportunity to leverage existing investments. IBM Workplace Web Content Management is now fully integrated into IBM WebSphere Portal and includes a new and simpler installation, improved integration with WebSphere Portal, enterprise user management and directory support enabling single sign-on across Workplace systems, and an enhanced, system-wide user interface (UI) rewritten as a WebSphere Portal application.

It manages the following:

- Content and design of pages in a portal or Web site.
- Framework and navigation of a portal or Web site.
- Creation, editing, approval, and publication process of a Portal or Web site's content.

The software comes in two varieties:

- IBM Workplace Web Content Management is appropriate for large and very large organizations.
- IBM Workplace Web Content Management, Standard Edition is an entry level offering for departmental or small and medium-sized organizations.

1.3.2 Understanding the versions and the positioning

Customers who are new to IBM Workplace Web Content Management might have questions about different versions or editions of the product. For example, what, if any, are the differences between the edition known as Web Content Management 5.1 and IBM Workplace Web Content Management 2.5?

IBM Workplace Web Content Management directly supports the vision of IBM Workplace — one vision — to provide the software which is based on a common set of services, based upon industry standard. Accordingly, IBM Workplace Web Content Management provides you with the same solution, if you are a Portal customer, a Workplace customer or a stand-alone customer looking for the most effective Web content management.

The three different approaches are as follows:

- If you are a *WebSphere Portal* customer, you use the embedded version of IBM Workplace Web Content Management 5.1. (WebSphere Portal includes a limited license. Refer to 1.6, “Licensing” on page 22 for further details on licensing.)
- If you are a customer of *Workplace Collaboration Services 2.5*, you get access to IBM Workplace Web Content Management 2.5 as one of services running on top of the core Workplace Collaboration Services foundation.
- Finally, if you are a stand-alone user who is interested in IBM Workspace Web Content Management (as a stand-alone Web content management tool), you have access to the stand-alone version.

Archived

1.3.3 Benefits of using IBM Workplace Web Content Management

IBM Workplace Web Content Management speeds Web content publication by removing its most common bottleneck: the overworked IT gatekeeper. It places end-to-end Web content management in the hands of content experts. This enables information updates that happen in minutes, not days. IBM Workplace Web Content Management is for customers who need to keep internet, intranet and extranet site information up-to-date, accurate, and under control. The product delivers faster time to market, lower costs of doing business, and a competitive advantage by consistently delivering personalized Web pages to site constituents so they get the right information, in the right context, at the right time. IBM Workplace Web Content Management integrates people with the information and business processes they need in a single secure, reliable, dynamic work environment for intranet, extranet, Internet and portal sites.

Using IBM Workplace Web Content Management provides the following functionality and benefits:

- Non-technical content owners can create and manage content, with an minimum on education effort
- Streamlines the Web content management process from content authoring, workflow, management, integration and delivery
- Publishes information about demand in minutes, not days for improved responsiveness to customers, partners, suppliers and employees
- Personalized content can be delivered to each user
- Provides an on-ramp to Lotus Workplace platform architecture
- IBM Workplace Web Content Management gives you a platform for high-value e-business applications
- The solution is scalable from pilot to enterprise
- Delivers faster implementation with component architecture that reuses components
- Reduces legal risks and associated costs
- Integrates with Lotus Domino Document Manager, Lotus Workflow™, IBM DB2 Content Manager, IBM WebSphere Portal
- The application programming interface (API) externalization feature gives organizations the ability to build solutions for migrating and importing from other Web content management products, doing advanced rendering and batch processing of content.
- Supports both J2EE™ and Domino

- Supports open standards to use and integrate smoothly your supported front-end, Web, existing, and backend systems into a unified solution, generally without re-engineering or expensive upgrades.
- TCO is reduced by enabling you to leverage existing infrastructures and skills
- Perhaps the most compelling reason to consider IBM Workplace Web Content Management is the ability to leverage the IBM portfolio. Benefits of IBM Workplace Web Content Management being positioned closely with Lotus Domino, WebSphere Portal and IBM DB2 are as follows:

  IBM is the only vendor to support WebSphere, WebSphere Portal, Lotus Domino, and DB2 Content Manager, producing the best total cost of ownership option, making this offering a great IBM middleware fit and an opportunity to leverage existing and future investments, especially for customers deploying IBM middleware.
- Workflow and versioning functionality can help you through the document review and approval process. Development features (which include templates to speed content creation) allow you to create Web pages that aggregate content from various sources and present it in a single, well-organized format. These pages can be customized for viewing on any device, including Web browsers, Pocket PC, Palm, or WAP/3G cell phones.

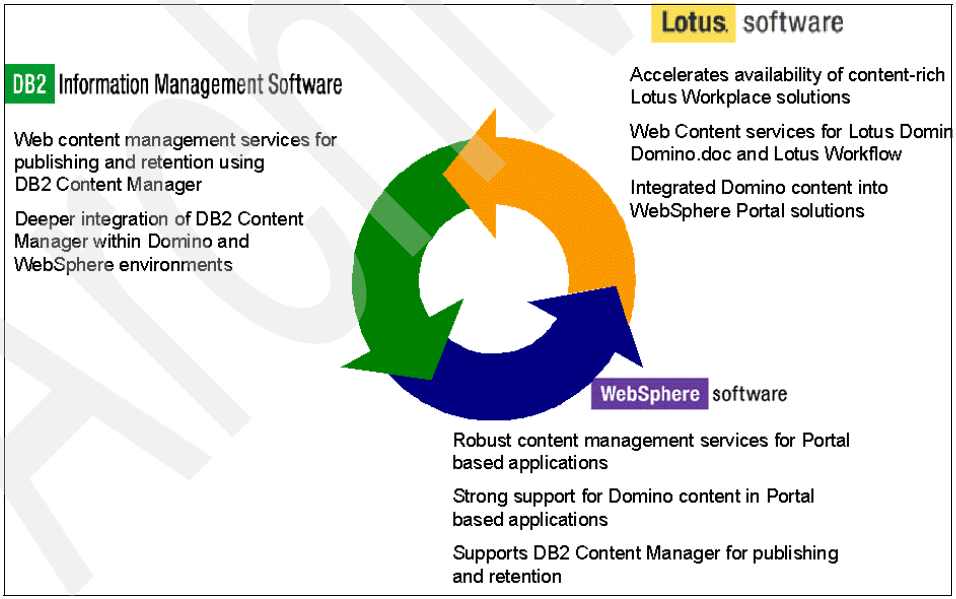


Figure 1-1 The functionality of Lotus, WebSphere, and DB2 software

1.3.4 The importance of data modeling

We wish to highlight the importance of data modeling, both as the primary prerequisite for designing and developing a new content management system with IBM Workplace Web Content Management, and as the means to organize information in a way that is readily accessible to users.

There are a number of different layers of data models that exist within IBM Workplace Web Content Management. These data layers provide the capability to share content within a Web site and across multiple Web sites and portals.

- Separating data from presentation

  The concept of separating content from it's presentation layout has been described in the beginning of this chapter. (See 1.2.2, “General concepts in Web content management” on page 8.) This allows for rendering a single piece of content in multiple ways. The content is the same, but the presentation can be adapted to best meet a user’s needs within the context of their specific role or preference. This approach also guaranties the consistent look and feel of a Web site. If the design changes, the content parts remain unaffected. This is important because content creators usually do not have significant knowledge of HTML.

- Information architecture: Separating data from the taxonomy

  Taxonomies help categorize and organize information for both the content creator and the content consumer so that content can be easily found. The separation of taxonomy and data also helps maximize re-use of information.

- Component object model

  IBM Workplace Web Content Management contains a powerful and flexible object model for modeling content and Web sites. At the core of the model you have a component. The component can be used to make up other components. These components can be re-used repeatedly within a site and shared with other sites. This reduces the time it takes to build a site, and ensures that the site has a consistent look and feel. Componentizing the site into separate objects allows different groups to be responsible for different elements of a Web site.

1.3.5 Highlights of new features since release 2.0

This section highlights the significant changes in the product from the 2.0 versions of IBM Workplace Web Content Management. Note that these features are introduced here but are also explained in much greater detail in subsequent chapters throughout the book.

New features include the following:

- IBM Workplace Web Content Management is now fully integrated with WebSphere Portal 5.1. The installation of WebSphere Portal comes along with a complete IBM Workplace Web Content Management installation. Only configuration steps are now required to deploy IBM Workplace Web Content Management.
- Templates: IBM Workplace Web Content Management now uses a portlet based user interface (the IBM Workplace Web Content Management Authoring Portlet) inside of the Portal or Workplace. BiDi characters are now supported in the IBM Workplace Web Content Management Authoring Portlet.
- Terminology Changes: Page Designs are now named Presentation Templates. Content Templates are now named Authoring Templates
- Dynamic presentation on stand-alone Web pages and within WebSphere Portals
- There are three Portlets used in the IBM Workplace Web Content Management environment:
  - Authoring Portlet to create and manage content
  - Local Rendering Portlet to show published content on the same server the authoring takes place
  - Remote Rendering Portlet to show published content on a different server
- An expanded set of worldwide language support for international organizations (including support for bi-directional content entry for Arabic and Hebrew languages).
- Users and Groups granted access to the IBM Workplace Web Content Management Authoring Portlet, IBM Workplace Web Content Management items and rendered Sites are now managed by WebSphere Member Manager.
- A Document Manager component has been added that can be used to reference WebSphere Portal Document Manager documents and display them as IBM Workplace Web Content Management components.
- Portal Search can be used to index and search IBM Workplace Web Content Management content delivered via the Rendering Portlet.
- API samples have been added to the product to assist developers use the IBM Workplace Web Content Management API.
- Documentation Improvements.
- Terminology Changes: Page Designs are now named Presentation Templates. Content Templates are now named Authoring Templates.

Archived

- Open standards (JSR 170 is a proposed API to standardize access to content) - to come
- Automated migration from version 2.0
- Support for HP/UX

1.4 Setting proper expectations about working with this product

This section looks at the big picture of what the IBM Workplace Web Content Management product can do for your organization. The success of an IBM Workplace Web Content Management project mainly depends on understanding the business value behind it and addressing the right resources for the key positions.

We provide some advice on how to get started, the roles of people that need to be involved and different key areas to focus on. At the end of this chapter we give you an overview what skills are needed for each of the roles involved, and which part of the book is most appropriate for these roles. You will find that this book is not only addressed to technical resources of an organization, but to many different stakeholders within a Web content management project.

IBM Workplace Web Content Management is a broad, powerful tool. Accordingly, it is important to think of it as a means for managing much more than a simple Web site, especially when you think of using it as a component within a dynamic workplace or portal.

Upon analyzing your existing business, you find many sections that are optimized to work together to fulfill the requirements. Simultaneously, you will also face business processes which lack of operational efficiency and that need improvements for more valuable information flow with customers, employees and partners. At this point you need to extract the key enterprise business objectives, barriers and communities that are involved. If you are considering using a Web content management system for only one department, you might think initially that this will not involve many communities. Analyzing this from a broader perspective however, you should consider different locations or organizational units and how these could be concerned or affected by future roll outs.

Single departments rarely remain isolated when it comes to sharing information. By eliminating those barriers and getting the right communities together, you can add significant business value and efficiency within your organization. These issues and processes are most frequently identified and refined by process

leaders working together with members of the management board. Changes in the way people share information - and ultimately how they perform tasks, often means changes in the way people work. End-to-end Web content management for Internet, intranet, extranet and portal sites can only be a success for an organization if the benefit and business value is obvious for all people involved.

Often Web content management projects are mistakenly declared as a pure IT project. This is an unfortunate and far too narrow view of this type of project. Keep in mind that a Web content management project is all about underlying information and providing a more effective way to manage and disseminate the information to customers, employees and any other consumer of this information. Accordingly, a Web content management project should be thought of in terms of an information project that consists of an information architecture as a foundation that is supported by an enterprise Web content management product. The Web content management system represents the key IT tool, because it enables the ability to manage and distribute the information. It should be clear however that business stakeholders, content owners, subject matter experts and process engineers will all need to be involved with the project.

When an appreciation and understanding has been established, that there must be numerous stakeholders involved, and that there is a clear business value associated with the proposed project, it is necessary to get proper sponsorship from the highest levels of management. Going forward, it will also be much easier get decisions from the management team, involve the subject matter authors at an early stage, motivate authors to deliver their information and announce clear requirements to the IT department to gain real business value.

1.4.1 The roles involved and their focus

Now that we understand the approach of using IBM Workplace Web Content Management to add business value, we examine the different roles in a typical IBM Workplace Web Content Management project and indicate the major functions.

Representative skills

We can define four primary groups that are needed in a typical IBM Workplace Web Content Management project. Depending on the specific role in a company it is possible that one and the same person can fill several positions at the same time.

- Business constituency

  This group consists typically of members who understand and oversee the the business vision. They know the line of business, can define the success criteria of the project and can help to define the underlying information architecture. In terms of business process, they can identify where the current

barriers exist that are to overcome with this project Typical members of this group include: members of the management board, a project sponsor, business owners, process leaders.

- Technical constituency

  This group represents the IT department or a specialized service provider. The technical business layer knows the existing technical architecture and understanding the technical requirements.

- Content constituency

  This group represents the subject matter experts, authors, content owners and the target group for whom the information is intended. Members of this group have a deep knowledge about existing application resources and content that needs to be integrated.

- Layout constituency

  This group is represented by designers and Web developers. Member of this group specialize in presenting information in an easy to consume way within a Web environment, while satisfying both the technical and corporate design requirements.

Note that while we have identified four typical groups of skill sets associated with a Web content management project, there should be frequent interaction between the groups and to some extent, there will be logical overlap between responsibilities. As shown in Figure 1-2 on page 19, the business, technical and content constituency need to interact with each other to be able to fulfill their roles and to pass on valuable information. Additionally, the design/ layout constituency needs to interact with all groups to support requirements that result from each group.

Archived

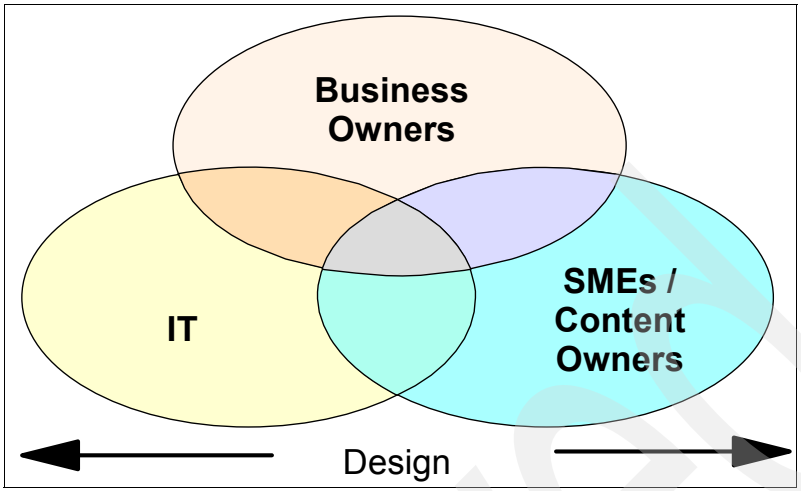


Figure 1-2 Interaction between the constituencies

Typical responsibilities within the different roles

The following section helps to identify typical tasks performed by members of different teams.

- Members of the management board and executive team perform the following tasks:
  - Announce the key business objectives
  - Set the business context
  - Define the project sponsor
  - Authorize the process leaders
  - Set the project budgets and timeline
- The project sponsor (part of the organization that is driving the project) performs the following tasks:
  - Work with business owners and content owners to define the information architecture
  - Define key concepts and terminology
  - Steer the decision process
  - Manage the project
- Process leaders perform the following tasks:
  - Manage the project management
  - Define the resources that are required

Archived

  - Work with business owners and content owners to define taxonomies that categorize and organize information for both the content creator and the content consumer. These tasks are closely related to the underlying information architecture.
- IT staff performs the following tasks:
  - The IT skill requirements of an IBM Workplace Web Content Management project requires medium to extensive knowledge of WebSphere Portal, IBM Workplace Collaboration Services, IBM Workplace Web Content Management and relevant Java technology. It is often reasonable to consider outsourcing the required IT infrastructure requirements to a specialized service provider. Core competencies required for the IT departments include the IT project management, configuration and management of the infrastructure and technical integration within the organization's environment.
  - Plan the architecture and deployment
  - Install and configure
  - Develop
  - Customize
- The design team performs the following tasks:
  - Help to create the corporate design
  - Work with IT team to technically implement the corporate design
  - Help to create the authors template environment
- Subject matter experts (SMEs) and authors perform the following tasks:
  - Provide content
  - Define the taxonomy
  - Define the content structure
  - Provide template support
- Content owners perform the following tasks:
  - Write content
  - Provide content quality assurance
  - Provide usability feedback
- The target group (users) performs the following tasks:
  - Address business needs
  - Provide feedback

Archived

1.5 How people use IBM Workplace Web Content Management

IBM Workplace Web Content Management rationalizes the entire process of publishing documents and content to a portal or Web site, starting at content authoring, to process through a workflow, to assure it gets published in the desired way at all target places. When a user updates a document, that update is made in all the appropriate places on the Web site, that could be an Internet, intranet and extranet. Finally, IBM Workplace Web Content Management works hand-in-hand with Portal to ensure that users see up-to-date information that is relevant to them, based on the user's role in the company and their access rights to specific information. This solution of dynamic Web content management can be used in a variety of different ways. In this following section, we highlight different solutions to increase the awareness of different possible usages.

Tool for managing large amounts of content

The most obvious way to use IBM Workplace Web Content Management is to maintain and publish content to a public Web site, such as `http://www.ibm.com`. In many cases, however, we also face areas where the content is only accessible for an authorized group of users. This content is restricted by reader rules that require authentication through a user name and password. The dynamic navigation is not visible for those who do not have rights to see the content.

Using within the context of Enterprise Content Management

Now let us get one step further and add in the described Web site enterprise content coming from existing non-Web application. This content can come from enterprise resource planning tools, document management systems such as DB2 Document Manager, databases applications such as an HR database, or finally, a procurement or groupware application. This common thread within this content is that it was never originally designed to be published within a Web environment. Furthermore, people in the enterprise or organization might not be aware that working with this content also impacts the content creation for the Web. IBM Workplace Web Content Management addresses the issue of handling content not originally designed for Web publication by enabling 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. The combination of IBM Workplace Web Content Management and IBM DB2 Content Manager V8 leverages the possibilities of Enterprise Content Management. For example a company might wish to offer their authorized customers the opportunity to view the status of their existing orders. This information is stored in various enterprise applications. IBM Workplace Web Content Management extracts the content,

publishes it as Web content using the correct Web design, while also controlling the required security policies for such delicate content.

As part of a Web application based on WebSphere Portal

Finally we like examine IBM Workplace Web Content Management as part of a powerful Web application. As we discussed earlier, we can use it to create stand-alone Web sites and publish content into WebSphere Portal. The WebSphere Portal framework provides us a platform to create world class Web applications. A complex product catalogue could be managed by a portlet application, and as a result, surface the content for specific product information out of our IBM Workplace Web Content Management system. We will take a closer look at this in Chapter 9, “Advanced integration with WebSphere Portal” on page 403.

1.6 Licensing

There are several ways to acquire IBM Workplace Web Content Management licences:

- By purchasing a IBM Workplace Web Content Management or IBM Workplace Web Content Management Standard Edition licence
- Though the entitlement in WebSphere Portal Enable or Extend
- As a part of a IBM Workplace Collaboration Services licence
- As a part of a software bundle.

IBM Workplace Web Content Management Stand-alone License

The IBM Workplace Web Content Management Products are licensed on a per-CPU basis and are available in two licensing variants:

- IBM Workplace Web Content Management 2.5 (full version)
- IBM Workplace Web Content Management, Standard Edition 2.5 (SMB/departmental)

The full product is intended for creating stand-alone Web sites, delivering content to WebSphere Portal, or a combination of the two. Therefore you would buy a licence for this product if you need to create a Web site or to supplement your WebSphere Portal entitlement licences (see below).

IBM Workplace Web Content Management, Standard Edition is an entry-level offering for small to medium-sized businesses with the same feature set provided

in the full product, however it has important licence restrictions. It is limited to 20 publishers in total (defined as any user who creates, edits, or reviews content), 4 CPUs maximum, a single server for hosting the live site (no clustering) and cannot be mixed with other IBM Workplace Web Content Management licenses (for example, WebSphere Portal entitlement licences or Workplace Collaboration Services). You cannot use it if you own WebSphere Portal Enable or Extend.

Both the Standard Edition and full version of IBM Workplace Web Content Management are available on either Domino or Java code The Java edition includes a runtime licence for WebSphere Portal, for the sole purpose of using IBM Workplace Web Content Management (from a technical perspective the Portal code is required for the UI, user management, and so forth.). Content can be published either to WebSphere Portal (if a Portal licence is owned) or to a non-Portal site (via the publishing servlet). Similarly the Domino edition includes a runtime licence for Domino Application Server.

WebSphere Portal Entitlement

Customers with WebSphere Portal Enable or WebSphere Portal Extend for Multiplattforms receive one or more restricted license entitlements to IBM Workplace Web Content Management, depending on how many Portal CPUs they have purchased. This limited license entitles customers to the Java edition only, and limits the use of IBM Workplace Web Content Management to *only within the Portal environment* (customers cannot publish to non-portal Web sites):

Customers receive 1 limited IBM Workplace Web Content Management entitlement for every six WebSphere Portal Enable or Extend CPUs purchased (not WebSphere Portal Express). For example, customers receive the IBM Workplace Web Content Management license as follows:

- 1 IBM Workplace Web Content Management CPU entitlement for the first through the sixth CPU of Portal
- 1 IBM Workplace Web Content Management CPU entitlement for seventh through twelfth CPU of Portal
- 1 IBM Workplace Web Content Management CPU entitlement for thirteenth through the eighteenth CPU of Portal
- 1 IBM Workplace Web Content Management CPU entitlement for each further sixth CPU of Portal (nineteenth, twenty-fifth, thirty-first, and so forth.)

The intention of the bundled licences is to meet the *basic scalability* Web content management capabilities required for a production portal system. For some portal deployments, this basic scalability will be all that is needed- an internal portal where most of the content is provided by portlets linking to enterprise applications could be an example of this. Many other WebSphere Portal

deployments (for example, content-oriented intranets and Web sites) require additional IBM Workplace Web Content Management licenses.

Portal customers who require additional CPUs for their Portal deployment, who wish to publish to non-portal sites or who require the Domino edition must purchase additional full IBM Workplace Web Content Management 2.5 licences. Customers who buy additional IBM Workplace Web Content Management processors for their WebSphere Portal 5.1 environment should purchase the full IBM Workplace Web Content Management product (it's the proof of entitlement), and install using the IBM Workplace Web Content Management code on the WebSphere Portal 5.1 media.

Portal Extend and Enable customers are not eligible to purchase IBM Workplace Web Content Management Standard Edition (SMB offering).

IBM Workplace Collaboration Services

Customers with IBM Workplace Collaboration Services are entitled to deploy IBM Workplace Web Content Management on each Workplace Collaboration Services CPU (IBM Workplace Web Content Management must be installed on the same server as IBM Workplace Collaboration Services). You would typically use this licence to publish content as part of a Workplace Collaboration Services application; however you might want to mix it with less expensive stand-alone IBM Workplace Web Content Management licences (for example, for a separate Authoring server).

Software bundles

IBM Workplace Web Content Management is also available in a number of software bundles from IBM. Many of these are licensed on a per-user basis, so are only suitable for intranets where the number of users is known. Regular CPU-based licences would be required for public Web sites.

Archived

Summary and more information

Table 1-1 Overview of licensing options

Offering	License	Trade-Ups
IBM Workplace Web Content Management (full version)	No license restrictions	Not Applicable
IBM Workplace Web Content Management, Standard Edition	► Limit to four CPUs ► Limit 20 publishers *total* (defined as anyone who touches the content: creates, edits, reviews) ► Limit *one* live server (no clustering) ► Can only be used with other Standard Edition licenses (no mixing with any other IBM Workplace Web Content Management licenses or entitlements, for example, WebSphere Portal Enable or Extend or Workplace Collaboration Services)	IBM Workplace Web Content Management, full version If a company wants to increase publishers, CPUs or live servers, they must trade-up to full IBM Workplace Web Content Management license and pay the difference
Workplace Collaboration Services	One licence of IBM Workplace Web Content Management for every Workplace Collaboration Services CPU purchased. Must be installed on the same server as Workplace Collaboration Services.	This is not a trade-up. However, if an organization needs additional IBM Workplace Web Content Management only servers and does not wish to purchase the full Workplace Collaboration Services licence (for example, for authoring, staging, and so forth.) they can purchase IBM Workplace Web Content Management, full version

Archived

Note: All the information in this chapter is correct at time of going to press. However, information is subject to change. You can find the latest information about licensing at the following:

- Current details on licence terms and conditions for all IBM Software products at:

  http://www.ibm.com/software/sla/

- Current information about pricing of IBM software in the IBM Software Catalogue at:

  http://www.ibm.com/software/info/ecatalog/

- IBM Workplace Web Content Management at:

  http://www.ibm.com/software/workplace/webcontentmanagement

- IBM WebSphere Portal at:

  http://www.ibm.com/software/websphere/portal

1.7 Overview of chapters in this book

Now that we have reviewed the typical roles and responsibilities as they relate to a typical IBM Workplace Web Content Management project, we can more specifically address the chapters of this book to different subsets of readers. This will help readers who have different roles within an IBM Workplace Web Content Management to understand which parts are most applicable to their role. (For a review of roles, see “Typical responsibilities within the different roles” on page 19)

This redbook is divided into four parts:

Part I: Introducing IBM Workplace Web Content Management

The current chapter provides an introduction to general concepts of Web content management and provides an executive overview of the IBM Workplace Web Content Management product. Additionally, it provides an introduction of the benefits, the expectations, and the success factors when planning an IBM Workplace Web Content Management project. It provides a context to better understand how IBM Workplace Web Content Management fits into the overall IBM Workplace and Portal strategy, while also providing an overview of the typical roles and responsibilities associated with a typical IBM Workplace Web Content Management project. This chapter is intended for executives and business owners looking to better understand IBM Workplace Web Content Management, its value, positioning and overview of features.

Part 2: Deployment planning and installation

Part 2 contains the following chapters:

- Chapter 2, “Product architecture and deployment planning” on page 33, provides an overview of the architecture of IBM Workplace Web Content Management 5.1 and discusses various techniques for implementing IBM Workplace Web Content Management within your infrastructure. Key concepts covered in this chapter include:
  - IBM Workplace Web Content Management 5.1 Product Architecture
  - IBM Workplace Web Content Management 5.1 content delivery techniques
  - Recommended environments for IBM Workplace Web Content Management 5.1 installations
  - Sample Infrastructure Architectures for IBM Workplace Web Content Management 5.1

  This chapter is intended for IT members looking to install and configure IBM Workplace Web Content Management.
- Chapter 3, “Installing and configuring IBM Workplace Web Content Management” on page 69 provides an overview of the process for installation and configuration of IBM Workplace Web Content Management 5.1.

  This chapter is intended for IT members looking to install and configure IBM Workplace Web Content Management
- Chapter 4, “Overview of IBM Workplace Web Content Management 2.5” on page 125 discusses the IBM Workplace Web Content Management 2.5 edition, highlighting its positioning, its target audience, and the subtle differences in the installation process, and its functionality and features.

  This chapter is intended for IT members looking to install and configure IBM Workplace Web Content Management.

Part 3: Key concepts, information architecture, site design, and reference implementations

Part 3 consists of the following chapters:

- Chapter 5, “Key concepts and terminology” on page 141 addresses the core concepts of IBM Workplace Web Content Management. We approach this topic by reviewing the separation of presentation and content and emphasizing the importance of complementation within page design. When this foundation has been established, we describe the key items, components, recommending important considerations for reuse and extensibility.

Archived

To provide a helpful context for the terminology and concepts, we use an example from the **River Bend Coffee and** Tea Company Web site (referred to in this chapter as **River Bend** Web site). You can see the site in Figure 7-1 on page 255. All concepts and terminology are shown using elements of this Web site. This chapter is intended for IT Members, content owners, designers and business owners wishing to understand the key concepts and terminology used with IBM Workplace Web Content Management.

- Chapter 6, “Information architecture and site design” on page 199 addresses key principles of information architecture and provides some recommended best practices to follow when structuring, designing, and implementing your site. It explains why good IA accomplishes an efficient Web site and also warns of the results of poor information architecture. Next we step by step develop the key considerations and the decision process starting at the business objectives we defined as success criteria. Followed by understanding the target group you like to reach, by evaluating a user's experience when browsing our reference site. We then approach the concept of Web site design by reviewing their fundamental principles and emphasize key components and their role in site design. Finally we close this chapter with considerations for the content authoring process to create a good experience. This chapter is addressed to the business owners, content authors and subject matter experts, and finally, the members of the design and layout constituency.
- Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253 describes a step by step guide on how to create a Web Site using IBM Workplace Web Content Management 5.1. The site created for the River Bend Coffee and Tea Company is referred to as River Bend Web site and contains content that describes the company’s business and provides information about its products. This chapter is addressed to the people who are new to building sites with IBM Workplace Web Content Management. It is very thorough, so it might be suitable for business owners, content authors and subject matter experts, and finally, the members of the design and layout constituency.
- Chapter 8, “Caching and pre-rendering considerations” on page 347 discusses performance considerations in IBM Workplace Web Content Management through the use of pre-rendering and caching strategies within IBM Workplace Web Content Management. More specifically, it discusses the following strategies:
  - 8.1.1, “Caching options” on page 349
  - 8.1.2, “Pre-rendering options” on page 351
  - 8.10, “WebSphere Dynamic Cache Service” on page 380

  This chapter is intended for members of an IT Team, looking to better understand strategies through which to utilize the most efficient caching method.

- Chapter 9, “Advanced integration with WebSphere Portal” on page 403 builds upon the basic “How to Build it” chapter, and takes the reader to the next step. It shows you how to build more tightly integrated IBM Workplace Web Content Management and WebSphere Portal solutions. It takes a phased approach to introduce you to the concepts and implementation. Step-by-step we will extend our solution and show you how to build it. Along the way we will explain the underlying Portal concepts and discuss the pros and cons of the outlined approaches. This chapter is addressed to developers and team members who have some experience with IBM Workplace Web Content Management and IBM WebSphere Portal.
- Chapter 10, “Search functionality for IBM Workplace Web Content Management” on page 429 reviews several ways you can allow site visitors to search for content stored in your IBM Workplace Web Content Management content repository. This chapter discusses the following topics:
  - IBM Workplace Web Content Management features for searching the IBM Workplace Web Content Management content repository
  - WebSphere Portal Search Engine features for searching the IBM Workplace Web Content Management content repository and how to use a custom approach for more effectively crawling, searching and displaying the results contained within IBM Workplace Web Content Management
  - Alternate techniques and third-party solutions to consider for searching the IBM Workplace Web Content Management content repository.

  This chapter is intended for members of an IT team who are looking to better understand strategies through which to configure and utilize the most efficient searching method.
- Chapter 11, “Personalization rules for IBM Workplace Web Content Management” on page 443 discusses how rule-based content delivery is implemented through a feature known as Personalization. Personalization lets Web sites customize their content automatically for each user. Personalization can recognize a specific user based on a profile, or determine characteristics of a user based on previous purchases, products or pages viewed, and so forth. Personalization then selects content that is appropriate for that profile. If a person has a high salary range, Personalization can be configured to retrieve information about a commercial Web site's premium products. The page is assembled with the proper personalized information, and the user sees her personalized page.
- Chapter 12, “Working with the application programming interface” on page 469 provides details on the IBM Workplace Web Content Management API. The goal of the chapter is to provide you with the ability to create custom Portlets and JavaServer™ Pages™ using IBM Workplace Web Content Management functionality. This chapter is addressed to developers.

Archived

- Chapter 13, “Advanced topics and examples with the API” on page 533provides advanced examples how to use the IBM Workplace Web Content Management API to create custom portlets. The chapter includes detailed example how to install and configure the Customizable Template Portlet (CTP). This chapter is addressed to developers.
- Chapter 14, “Migrating to IBM Workplace Web Content Management 5.1” on page 565 describes migrating a Web site from a previous version of IBM Workplace Web Content Management into IBM Workplace Web Content Management 5.1. This chapter is intended for users who have used previous versions of IBM Workplace Web Content Management, considering migrating in content from the earlier version to IBM Workplace Web Content Management 5.1.
- Chapter 15, “Migrating content into IBM Workplace Web Content Management” on page 599 describes how to migrating a Web site from another Web content management tool into IBM Workplace Web Content Management 5.1. For the purposes of this chapter, we will highlight the features and functionality of Vamosa’s plug-in for IBM Workplace Web Content Management. This chapter is intended for users who have created data in other Web content management systems, and are considering migrating in content from the outside version to IBM Workplace Web Content Management 5.1 using Vamosa’s tool.

Archived

Part 2

Deployment planning and installation

Archived

© Copyright IBM Corp. 2006. All rights reserved.

Archived

2

Product architecture and deployment planning

This chapter provides an overview of the architecture of IBM Workplace Web Content Management 5.1 and discusses various techniques for implementing IBM Workplace Web Content Management within your infrastructure. Prior to an in depth discussion of physical implementation architectures, this chapter looks at how IBM Workplace Web Content Management fits within the WebSphere Portal framework as well as the various techniques for displaying IBM Workplace Web Content Management-based content.

This chapter includes the following sections:

- IBM Workplace Web Content Management 5.1 Component Architecture
- Rendering IBM Workplace Web Content Management Content
- IBM Workplace Web Content Management environment types
- Sample physical architectures
- Example infrastructure

© Copyright IBM Corp. 2006. All rights reserved.

2.1 IBM Workplace Web Content Management 5.1 Component Architecture

The IBM Workplace Web Content Management 5.1 application is fully integrated with WebSphere Portal 5.1. As such, all required components of the IBM Workplace Web Content Management application are installed with WebSphere Portal 5.1. The following diagram provides a high level overview of where IBM Workplace Web Content Management fits into the WebSphere Portal infrastructure.

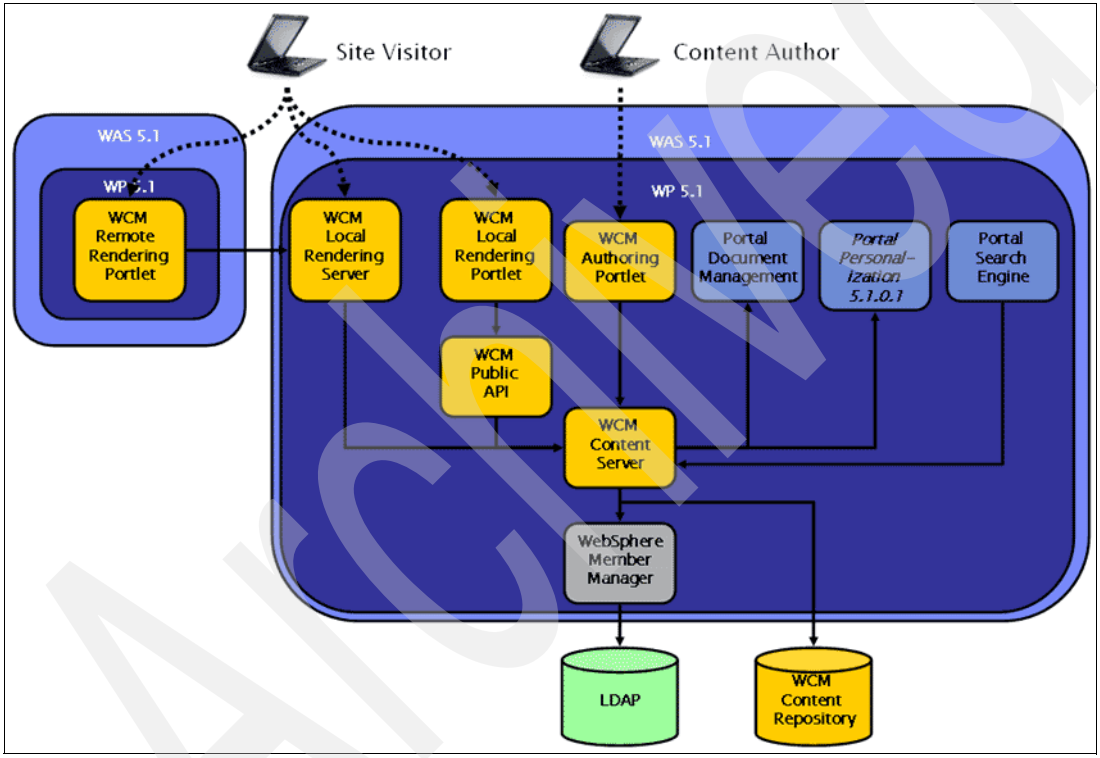


Figure 2-1 IBM Workplace Web Content Management architectural overview

Archived

2.1.1 Key components

This section defines the key components of IBM Workplace Web Content Management 5.1.

IBM Workplace Web Content Management Content Server

The IBM Workplace Web Content Management Content Server is the *core* of the IBM Workplace Web Content Management application. All requests for content are ultimately processed by the Content Server.

Unlike earlier versions of IBM Workplace Web Content Management, the IBM Workplace Web Content Management 5,1 Content Server leverages WebSphere Member Manager as its user repository. This repository is used for both authentication as well as determining group membership for authenticated users.

IBM Workplace Web Content Management Content Repository

When you first enable IBM Workplace Web Content Management, it uses WebSphere Portal's embedded Cloudscape™ database as its content repository. This is true even if you have migrated your WebSphere Portal database to another database platform.

If desired, you can switch your IBM Workplace Web Content Management repository to any of the supported databases. IBM Workplace Web Content Management 5.1 application data and resources can be stored in the following databases:

- DB2 8.1 with FixPack 6 or 6a (Enterprise and Workgroup Editions)
- Oracle 9.2.0.4
- IBM Content Manager 8.2 FixPack 7
  (IBM Information Integrator for Content 8.2 FixPack 3 must also be installed)
- Cloudscape 5.1.60.12
- Microsoft® SQL Server 2000 SP3a
- UDB for z/OS7.1

Notice that one of your options for content storage is IBM Content Manager. If your organization is working towards an enterprise content management solution it is worth noting that IBM Workplace Web Content Management can use that same enterprise content management solution as its content repository.

Archived

Note: For details on using IBM Content Manager as the repository, refer to *IBM Workplace Web Content Management Java Edition V2 and IBM DB2 Content Manager V8 Working Together*, SG24-6368, which is available at:

http://www.redbooks.ibm.com/abstracts/sg246368.html?Open

Tip: Even if you choose not to store your content in IBM Content Manager you can still reference items stored in IBM Content Manager from within IBM Workplace Web Content Management.

As this list of supported databases is subject to change, refer to the WebSphere Portal 5.1 Information Center for details on the latest supported versions and platforms, which is available at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html

In addition, refer to each product's documentation for details on the specific hardware and software requirements for these databases.

IBM Workplace Web Content Management Authoring Portlet

Beginning with IBM Workplace Web Content Management 5.1 the user interface for working with IBM Workplace Web Content Management is provided via a Portlet running within WebSphere Portal. In prior versions of IBM Workplace Web Content Management there was a stand-alone browser environment for these activities. The new Portlet provides significantly more functionality than the old user interface.

While referred to as the Authoring Portlet, this Portlet is (more accurately) your sole graphical user interface for interacting with IBM Workplace Web Content Management. Content creators use the portlet to author content. Content approvers use this same portlet for managing content. Developers use this portlet to create technical assets (for example, Presentation Templates, Workflows, Syndication components, and so forth).

The Authoring Portlet allows for very granular user interface security controls. These controls did not exist in prior user interfaces. (See 3.2, “Installing IBM Workplace Web Content Management” on page 73 for more details on configuring and securing the Authoring Portlet.)

Restriction: The Authoring Portlet must be running on the IBM Workplace Web Content Management server itself. Unlike the Remote Rendering Portlet, there is no version of the Authoring Portlet that supports accessing IBM Workplace Web Content Management on a remote WebSphere Portal server.

IBM Workplace Web Content Management Local Rendering Server Servlet

The Local Rendering Servlet is used to deliver Web content outside of a WebSphere Portal environment. You would use this servlet when you need to deliver a traditional stand-alone Web site. Site visitors can access content by requesting the HTML directly from the IBM Workplace Web Content Management Local Rendering Servlet.

It is important to note that even if you only use IBM Workplace Web Content Management to deliver stand-alone (non-portal) Web sites, the core IBM Workplace Web Content Management application always runs on WebSphere Portal and Lotus Workplace. There is no version of IBM Workplace Web Content Management that can run without one of these.

IBM Workplace Web Content Management Rendering Portlets

Traditionally, one of the more difficult parts of integrating an IBM Workplace Web Content Management solution with WebSphere Portal was displaying the content itself within WebSphere Portal. IBM Workplace Web Content Management eliminates this hurdle by providing two Portlets that can display your IBM Workplace Web Content Management content in WebSphere Portal. These portlets require no Java coding - only simple configuration by the portal administrator.

IBM Workplace Web Content Management Local Rendering Portlet

The Local Rendering Portlet is used to display IBM Workplace Web Content Management content within a Portlet. This portlet can only be used when deployed to a WebSphere Portal Server that also has a live instance of IBM Workplace Web Content Management.

The Local Rendering Portlet relies on the IBM Workplace Web Content Management API to interact with the IBM Workplace Web Content Management Content Server.

Archived

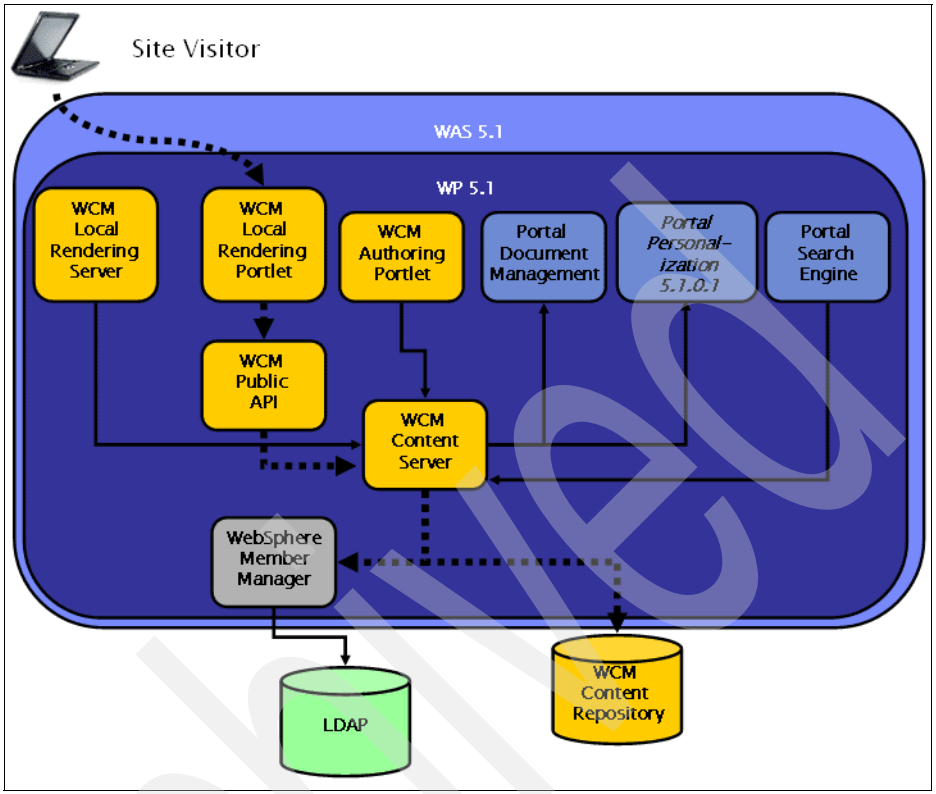


Figure 2-2 Basic flow of activity when using the Local Rendering Portlet

Workplace Web Content Management Remote Rendering Portlet

The Remote Rendering Portlet is used to display IBM Workplace Web Content Management content within a Portlet. This portlet should only be used when deployed to a WebSphere Portal Server that does not have a live instance of Web Content Management.

Unlike the Local Rendering Portlet, the Remote Rendering Portlet uses an HTTP connection to interact with a remote IBM Workplace Web Content Management Content Server at runtime. The configuration of this portlet contains some additional fields to facilitate this type of connection.

Archived

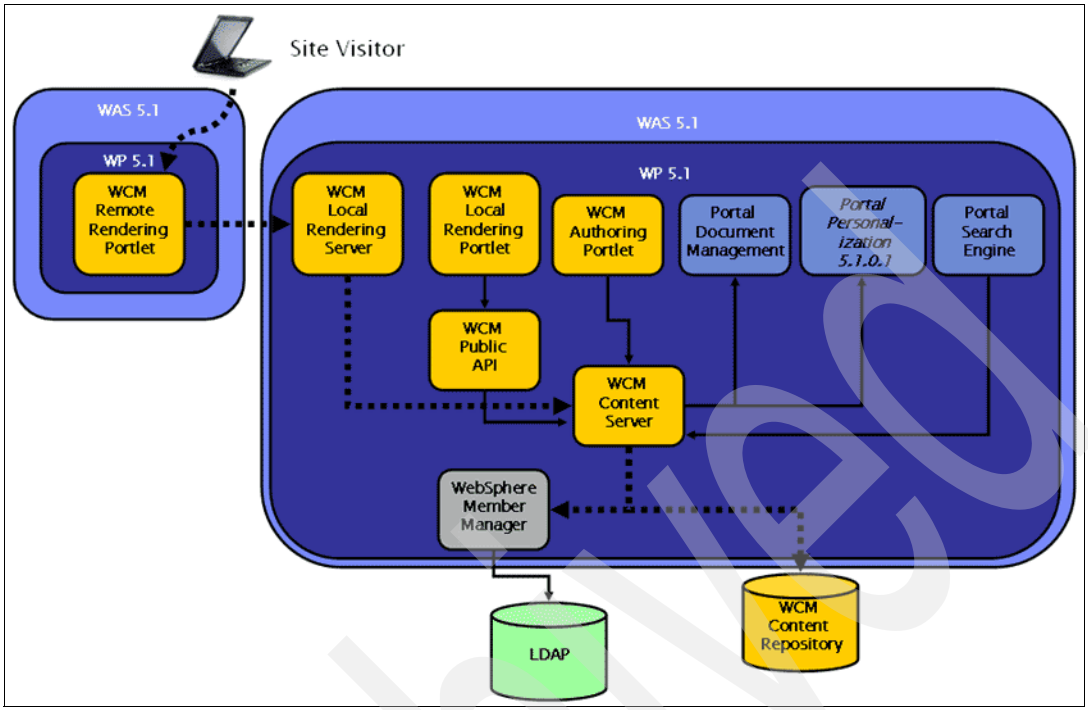


Figure 2-3 Basic flow of activity when using the Remote Rendering Portlet

Tip: Basic performance testing in IBM Workplace Web Content Management 5.1 has shown that the Remote Rendering Portlet performs marginally better than the Local Rendering Portlet. However, it is important to be aware of your infrastructure when using the Remote Rendering Portlet. firewalls, Proxies, and other network issues can have a dramatic effect on the performance of the Remote Rendering Portlet.

IBM Workplace Web Content Management Public API

While IBM Workplace Web Content Management provides a solid set of Web content management capabilities *out-of-the-box*, there are times when the standard capabilities of IBM Workplace Web Content Management do not meet your exact needs. For situations such as this IBM Workplace Web Content Management provides a Java API.

It is important to note that the IBM Workplace Web Content Management API does not expose all the capabilities of the IBM Workplace Web Content Management application. We recommended you review the Javadoc™ provided with the API to review the features and functions that are exposed. You can find the Javadoc at:

```
<WAS_Root>\installedApps\<cell_name>\wcm.ear\ilwwcm.war\webinterface\
api-javadoc
```

In this path name, *<WAS_Root>* is the root directory for WebSphere Application Server (for example, D:\WebSphere\AppServer) and *<cell_name>* is the name of the Cell on this WebSphere Application Server instance (for example, wcm01).

For example, these files were found on our wcm01.cam.itso.ibm.com server at:

```
D:\WebSphere\AppServer\installedApps\wcm01\wcm.ear\ilwwcm.war\webinterface\
api-javadoc
```

Restriction: The IBM Workplace Web Content Management API only supports Java code running in the same Java Virtual Machine as IBM Workplace Web Content Management itself. If you wish to access the IBM Workplace Web Content Management API remotely you will need to build some type of service layer on the IBM Workplace Web Content Management server to broker your requests. The most common solution to this issue is to develop a Web Service to perform this function.

For an overview of the basic capabilities provided via the IBM Workplace Web Content Management API, see Chapter 12, “Working with the application programming interface” on page 469 and Chapter 13, “Advanced topics and examples with the API” on page 533.

IBM Workplace Web Content Management JavaServer Pages Tags

In addition to the Java API, IBM Workplace Web Content Management provides a JavaServer Pages Tag Library that you can use when developing Portlets and other J2EE applications. The tags in this library make it very easy to access your IBM Workplace Web Content Management content from a JSP™ page. These tags rely on the API for their functionality so they do not provide any capabilities beyond those of the API.

Restriction: Because the JSP Tags rely on the IBM Workplace Web Content Management API for functionality they must run in the same Java Virtual Machine as the IBM Workplace Web Content Management server.

For more details on the API and JSP Tags, see the IBM Workplace Web Content Management Information Center.

2.1.2 Changes from earlier versions

This section discusses those changes made to IBM Workplace Web Content Management 1.0 and higher and 2.0 and higher.

Embedded within WebSphere Portal and IBM Workplace Collaboration Services

Prior versions of IBM Workplace Web Content Management were provided as stand-alone Web applications. IBM delivered the application as an EAR file and it was installed directly onto WebSphere Application Server.

Starting with WebSphere Portal 5.0.2.2 and 5.1 and IBM Workplace Collaboration Services 2.5, IBM Workplace Web Content Management is no longer packaged in this format. Even if you only plan on creating non-portal Web sites with IBM Workplace Web Content Management you must still install WebSphere Portal (or Workplace Collaboration Services 2.5) in order to gain access to the Authoring Portlet.

When you purchase a license for IBM Workplace Web Content Management you are given a limited use license of WebSphere Portal (or Lotus Workplace) in order to run your IBM Workplace Web Content Management environment.

Multiple Rendering Portlets

Now that IBM Workplace Web Content Management is integrated with WebSphere Portal, IBM provides a new Local Rendering Portlet for accessing IBM Workplace Web Content Management content. The single portlet provided with early versions is now referred to as the Remote Rendering Portlet.

Remote Rendering Portlet

The Remote Rendering Portlet is the latest version of the IBM Workplace Web Content Management portlet that has been available since IBM Lotus Workplace Web Content Management 1.0. The Remote Rendering Portlet is intended to be run on any WebSphere Portal instance that does not have IBM Workplace Web Content Management enabled. This portlet utilizes a browser-like URL connection to talk to the IBM Workplace Web Content Management server. It does not leverage the IBM Workplace Web Content Management API to perform any of its work.

Local Rendering Portlet

The Local Rendering Portlet is a new portlet that started shipping with IBM Workplace Web Content Management 5.1. This portlet is only for use on a WebSphere Portal server that has IBM Workplace Web Content Management enabled. Unlike the Remote Rendering Portlet, this local version utilizes the IBM

Workplace Web Content Management API for it's interaction with IBM Workplace Web Content Management content.

User Repository

Versions 1 and later and 2.0 and later of IBM Workplace Web Content Management utilized their own proprietary user repository. While it was possible (via custom Java code) to utilize an LDAP to authenticate site visitors, the proprietary user repository was required for storing user information for anyone who needed to access the IBM Workplace Web Content Management User Interface (for example, content authors, content approvers, system administrators).

IBM Workplace Web Content Management 5.1 and 2.5 now leverage the LDAP and WebSphere Member Manager capabilities of WebSphere Portal and Lotus Workplace. This single repository is used as the authentication mechanism for all aspects of IBM Workplace Web Content Management.

Cloudscape versus File system for content repository

Earlier versions of IBM Workplace Web Content Management supported multiple types of content repositories such as DB2, Oracle, and DB2 Content Manager. These earlier versions also supported using the file system as a data store. This file system approach was convenient for prototypes or development work but were not appropriate for production environments.

Following the same approach as WebSphere Portal and IBM Workplace Collaboration Services, all portal-based versions of IBM Workplace Web Content Management (5.1 and 2.5) now use a Cloudscape database as their default content repository. File system storage is no longer supported.

If you have an older version of IBM Workplace Web Content Management that uses file system storage you will need to migrate the data as you upgrade to IBM Workplace Web Content Management 5.1 or 2.5.

Attention: The migration from a file system environment only supports migration to Cloudscape. If your ultimate content repository is another database such as DB2 or Oracle you will need to first migrate the file system content to Cloudscape and then to your final destination such as DB2 or Oracle.

2.2 Rendering IBM Workplace Web Content Management Content

One of the key benefits of IBM Workplace Web Content Management is the separation of content and structure. The raw content is not combined with any structure (for example, HTML tables) until a user requests the content. It is at this time that IBM Workplace Web Content Management renders the page by combining the content and presentation elements.

In IBM Workplace Web Content Management 5.1 there are several ways you can render content. In this section, we look at the various rendering options that are available in the current product.

2.2.1 An overview of the rendering options

At a high level, there are three basic ways that IBM Workplace Web Content Management can render content:

- Stand-alone Web site (via Local Rendering Server)
- WebSphere Portal page (via Local/Remote Rendering Portlets)
- Java API

A single installation of IBM Workplace Web Content Management can support any combination of these rendering techniques.

Rendering IBM Workplace Web Content Management content in a stand-alone environment

IBM Workplace Web Content Management provides the ability to generate non-portal Web sites via its Local Rendering Server. This scenario is commonly used in two situations:

- Your company does not currently use WebSphere Portal / Lotus Workplace as a delivery platform.
- You want to make IBM Workplace Web Content Management content available to a certain audience but do not want to add additional WebSphere Portal and Lotus Workplace servers to accommodate the incremental users.

In either of these scenarios you can use the IBM Workplace Web Content Management Rendering Server to serve content directly to a Web browser.

Archived

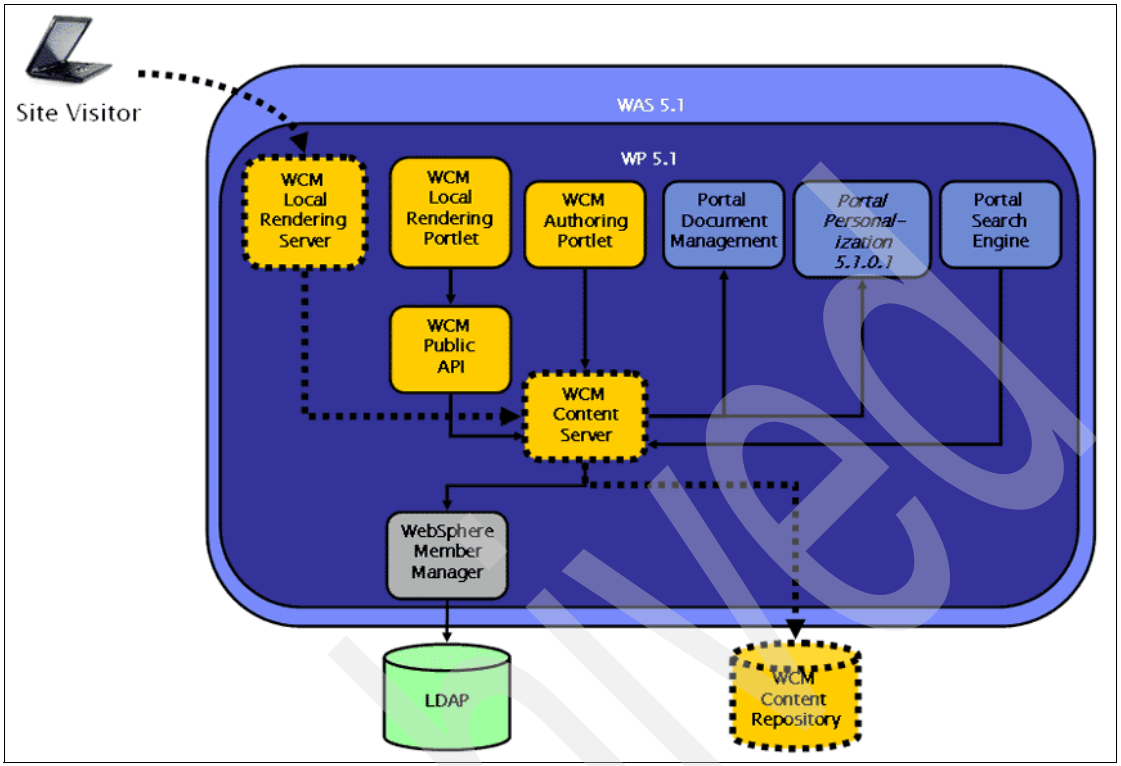


Figure 2-4 Using IBM Workplace Web Content Management to deliver non-Portal (stand-alone) content

Delivering content to a stand-alone Web site involves only a portion of the IBM Workplace Web Content Management software infrastructure. When user requests content via the IBM Workplace Web Content Management Rendering Server, that request is sent to the IBM Workplace Web Content Management Content Server for processing. The IBM Workplace Web Content Management Content Server will gather the appropriate assets from the IBM Workplace Web Content Management Content Repository and send the complete page back to the requestors browser.

If your content is secured, the IBM Workplace Web Content Management Content Server will also utilize WebSphere Member Manager to ensure that the authenticated user has been granted access to the requested content.

Static pre-rendering of content in IBM Workplace Web Content Management

The pre-rendering functionality offered in IBM Workplace Web Content Management allows a site to be pre-rendered, thus decreasing the load on the

server when delivering Web pages. This can significantly improve the performance of a site. There are two options available for pre-rendering:

1. Pre-render the site but still have IBM Workplace Web Content Management control the delivery of the Web site. (Basic Site Caching)
2. Pre-render the site and deploy the static HTML via a Web server. (Static Pre-Rendering)

We will discuss the first option when we look at the advantages of caching in Chapter 8, “Caching and pre-rendering considerations” on page 347.

The second option is used typically when the Web site is both static and anonymous. In this second scenario, you would pre-render the entire site and then move the pre-rendered files to an external HTTP server. This decreases the licensing required for your IBM Workplace Web Content Management infrastructure because IBM Workplace Web Content Management is not running on your production servers. However, it also eliminates any possibility for dynamic behavior. Think carefully before you utilize the static pre-rendering option. It can save you money in licenses, but also **significantly** reduces the IBM Workplace Web Content Management features available to your site.

Rendering IBM Workplace Web Content Management via WebSphere Portal

IBM provides a pair of portlets that can render IBM Workplace Web Content Management content within WebSphere Portal. The advantage of using these portlets is that you can render IBM Workplace Web Content Management content within WebSphere Portal without writing any Java code (a portlet). The IBM Workplace Web Content Management rendering portlets rely on the components built in IBM Workplace Web Content Management (for example, Presentation Templates, HTML components and so forth) to build the presentation. These portlets significantly decrease the time and effort needed to display IBM Workplace Web Content Management content within WebSphere Portal.

it is important to note that while the two rendering portlets work in a very similar fashion from a user interface perspective, the underlying technologies for these two portlets are very different. The Local Rendering Portlet utilizes the IBM Workplace Web Content Management API while the Remote Rendering Portlet communicates with the IBM Workplace Web Content Management Local Rendering Server (servlet) via a URL connection.

Refer to 2.2.1, “An overview of the rendering options” on page 43, for a description of the two portlets that IBM provides and where each is applicable for deployment.

IBM Workplace Web Content Management API

While IBM has certainly made accessing IBM Workplace Web Content Management simple via the rendering portlets and rendering server, there are times when neither of these processes provide you exactly what you need. In many of these cases the public API for IBM Workplace Web Content Management gives a developer the flexibility to create applications or portlets that more accurately meet with project requirements.

Restriction: The IBM Workplace Web Content Management API only supports Java code running in the same Java Virtual Machine as IBM Workplace Web Content Management itself. If you wish to access the IBM Workplace Web Content Management API remotely you will need to build some type of service layer on the IBM Workplace Web Content Management server to broker your requests. The most common solution to this issue is to develop a Web Service to perform this function.

2.2.2 Alternate Rendering Concepts

In many customer environments there are times when the standard rendering capabilities of the Local/Remote Rendering Portlets and IBM Workplace Web Content Management Rendering Server are not sufficient to meet a project's needs. In situations such as these there are certainly other options available to rendering content. In this section, we review two such techniques that are quite common.

XML and RSS

Really Simple Syndication (RSS) is a lightweight XML format designed for sharing headlines and other Web content. RSS has evolved into a popular means of sharing content between sites. RSS can also be the basis for additional content distribution services (a.k.a news feeds). There are many freely available RSS news readers that can scan and aggregate content from multiple RSS news feeds.

It is very easy to build an IBM Workplace Web Content Management Menu, Navigator, or Presentation Template that generates an RSS news feed or other XML-based output. An example of this process is outlined in *Lotus Workplace Web Content Management*, SG24-6309, which is available at:

http://www.redbooks.ibm.com/abstracts/sg246309.html?Open

Tip: At its core, RSS is just a specific XML format. You can extrapolate this concept into other XML-based delivery techniques.

WML and cHTML

Just with the previous RSS example, it is also possible to generate other markup languages such as WML or cHTML.

If you are delivering content via WebSphere Portal there are other ways to accommodate these types of markup languages, but if you are using the IBM Workplace Web Content Management Rendering Server to deliver content you could create unique presentations of the content for various presentation languages (for example, IBM Workplace Web Content Management, HTML, cHTML).

2.3 IBM Workplace Web Content Management environment types

In a typical IBM Workplace Web Content Management installation there are multiple physical IBM Workplace Web Content Management servers. In our experience, these servers end up performing one of the following roles within the IBM Workplace Web Content Management infrastructure:

- Development
- Authoring
- Staging
- Production (Live)

2.3.1 Development

This is where you develop and unit test the Web Content Management technical assets such as Presentation Templates, HTML Components, Menus, and Navigators. In addition, this server can be the first place you install patches and fixpacks to ensure they do not have a negative impact on your IBM Workplace Web Content Management servers.

Typically, you syndicate changes to the authoring environment, to roll out changes. These changes are, in turn, syndicated or replicated out to the staging and live environments after the appropriate testing.

Archived

Restriction: IBM Workplace Web Content Management does not currently support the notion of selective replication. In other words, you cannot tell IBM Workplace Web Content Management to only Syndicate design changes from the Development server to other servers in your environment. This limitation means you must be very careful when Syndicating content to from a Development server.

One approach is to set the Syndicator to Syndicate only Live content and delete all content (via the API) prior to Syndication.

Another similar approach is to configure Syndication the same way but have no Live content in the Development server.

Regardless of your preferred approach, be *very careful* when syndicating from Development to other servers.

2.3.2 Authoring

This is where content authors and approvers create, edit, preview, approve, and expire content. This environment can be distributed if required. For example, if you have offices across the globe, it might make sense to have authors work on a server closer to their physical location as opposed to a centralized authoring system.

If you choose to have a distributed Authoring environment it is important to make sure people on separate servers do not attempt to work on the same content simultaneously. In situations where distributed Authoring environments make sense you should secure your IBM Workplace Web Content Management environment such that people on each Authoring server can only modify a unique set of data.

Restriction: Due to a restriction in the IBM Workplace Web Content Management software you cannot author content from a server in a cluster. Remember this restriction when designing your environment!

2.3.3 Staging

This is sometime perceived as merely nice to have, but a staging environment is highly recommended as a safety net. A staging environment provides a chance to review changes in the context of your whole site before they are syndicated into the live environment. This can include previewing the Web content as well as integration to/from other applications. A staging environment is particularly useful if you are pushing out a large number changes simultaneously.

A staging environment can also tie into your test environment, depending on the relevant testing that is part of your Content Management and WebSphere Portal Development life cycles.

From a best practices perspective, *the Staging environment should use all functional elements found in the Production (Live) environment*. For example, if your Production environment contains a cluster for IBM Workplace Web Content Management servers so should your Staging environment. Designing your staging environment in this manner provides two potential benefits:

- Testing

  If you need to test a new configuration, install a patch/fix, and so forth, you need a place to test such changes prior to implementation on the production servers. If the Staging environment mirrors your Production environment you can confirm such changes in Staging.

- Performance testing

  In an ideal environment, Staging would be an exact duplicate of the Production environment. If this were the case you could easily do performance testing in the Staging environment. If the Staging environment contains all elements of Production but on a smaller scale you can still use this environment to run performance testing, but you will have to extrapolate the results, which could lead to incorrect performance expectations.

2.3.4 Production (live)

This environment provides the live site to Site Visitors and is tuned for fast rendering of Web pages and portlets. This environment is often clustered for both failover and performance. Live environments often include other performance-enhancing software and hardware to help with caching, load balancing, and failover.

Restriction: The Authoring Portlet does not function properly on an IBM Workplace Web Content Management server that is installed into a cluster. Authoring cannot be done in a clustered environment. Furthermore, in order to make configuration changes to production servers (some of which are done via the Authoring Portlet) you will have to remove the server from the cluster prior to making such changes.

2.3.5 Workflow versus syndication

With multiple servers in a typical IBM Workplace Web Content Management environment, it is necessary to have a mechanism that can synchronize data between multiple IBM Workplace Web Content Management servers. In addition,

it is a typical requirement to have content pass through some formal approval process before it can be viewed on the Web site. IBM Workplace Web Content Management provides solutions for both of these issues with Workflow and Syndication. New users of IBM Workplace Web Content Management frequently interchange these two terms as though they are one and the same. However, these two capabilities are complementary and you must make sure you understand the role of each within IBM Workplace Web Content Management.

Workflow

Workflow controls the creation, approval, and runtime (live) access to an item. A workflow consists one or more stages. All content in IBM Workplace Web Content Management must go through a workflow process - even it that workflow is a simple one step “express” approval.

> **Tip:** Content cannot be seen by site visitors at runtime (live) until the content is approved.

Syndication

Syndication is the method used by an IBM Workplace Web Content Management server to replicate data from one IBM Workplace Web Content Management server to another. Unlike Workflow, Syndication is not involved in the process of approving content. Syndication is only responsible for replication of IBM Workplace Web Content Management assets across multiple servers.

Figure 2-5 Item Gatherer select list when creating a Syndicator

When creating a Syndicator, IBM Workplace Web Content Management allows you two choices for the granularity of items to be syndicated:

- **All Items:** When configured with this setting, all elements of the IBM Workplace Web Content Management server are replicated to the other server - including draft and expired content. Typically Authoring / Development servers need access to all types of content for testing purposes. This setting is common when syndicating between distributed Authoring servers or between an Authoring server and Development server.
- **All Live Items:** As opposed to the `All Items` setting, this option only replicates live content. Live content is defined as content that has been approved via workflow (live) but not yet past its expiration date. Typically a production server would only receive live data - so this setting is common when syndicating from Authoring to Staging or Staging to Production.

Tip: Syndication will always syndicate copies of your technical assets (for example, Presentation Templates, Authoring Templates, Library Components, and so forth). By default there is no setting to stop this. You can configure IBM Workplace Web Content Management to workflow these items at which point they would follow the same rules as content.

How syndication works

To participate in syndication, define two servers as a pair that will syndicate. The syndicator is the server that contains the data that needs to be copied to another server. The subscriber is the server that retrieves the copies of the data from the syndicator. The servers do not need to be physically connected; they need only be network accessible.

The syndicator contains an item gatherer, which registers all the items that change. A change might mean an item is created, edited, or deleted; for example, new content is created, or existing content is edited or deleted.

Changes usually occur during peak periods on the development server. Syndication affects performance, and should only run during off-peak periods when no changes occur. However, you cannot schedule syndication to run at a specific time.

A task that runs every 30 seconds, by default, registers whether the item gatherer contains changed items. If the item gatherer does contain changed items when the task runs, syndication does not run. Changed items in the item gatherer indicate that the server is in a peak period of use. If the item gatherer does not contain changed items when the task runs, syndication runs. No changed items in the item gatherer indicate that the server is in an off-peak period.

During syndication, the syndicator sends the URLs of the changed items to the subscriber. The subscriber retrieves the items using the URLs and copies them to its data store. If the item already exists on the subscriber, the newest item, defined by the modification date, replaces the existing item. Items cannot be appended or merged.

Different servers might store data using different data storage types. For example, the development server might store data using a Cloudscape database, while the live server might store data using a DB2 database. IBM Workplace Web Content Management can syndicate data between servers that use different data storage types.

Archived

2.3.6 Putting the environments together

Based on the need for multiple IBM Workplace Web Content Management environments you will typically end up with multiple IBM Workplace Web Content Management servers in your infrastructure. Figure 2-6 shows what a typical combination of these environments might look like:

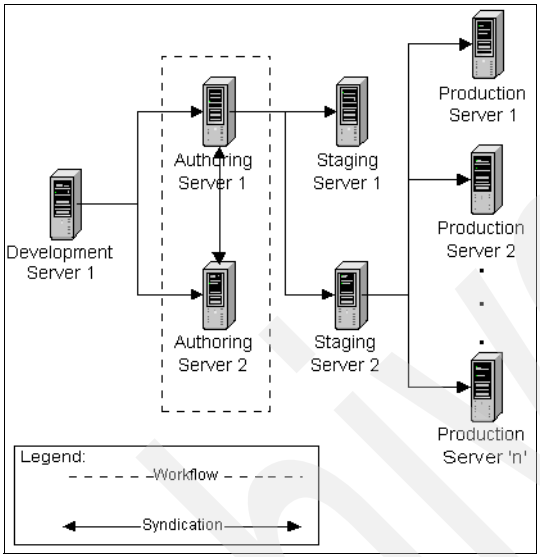


Figure 2-6 Typical configuration and content flow

A Development Server is set up to allow developers to create the technical assets (Presentation Templates, Workflows, and so forth) utilized within the IBM Workplace Web Content Management environment. These assets are **syndicated** to the Authoring Servers.

One or more Authoring Servers are utilized to create and approve content. The Authoring servers will contain content in a variety of stages. The IBM Workplace Web Content Management workflow processes take place on these servers. When content is approved, it is syndicated to the Staging Server(s).

The Staging servers should contain *only* Live (approved) content. This can be controlled via Syndication. When all content is reviewed and deemed ready for production, the content is syndicated to the Production Server(s).

The Production Server(s) are responsible for delivering content to Site Visitors at runtime.

Archived

2.4 Sample physical architectures

In this section, we look at some physical architectures that you can use to build the IBM Workplace Web Content Management infrastructure.

Attention: You can use the architectural designs described in this section to build out any of the environments described in the previous section. In practice, you combine some number of these architectures across the various environments. At the end of this chapter, we build a sample solution that pulls together all the ideas in this chapter.

In real-world installations of IBM Workplace Web Content Management there are a variety of common infrastructure designs in place. The primary differences in these designs are due to variations in several basic assumptions:

- Site size and complexity: For a relatively small Web site, it might not be necessary to have individual servers dedicated to all four types of IBM Workplace Web Content Management environments.
- Funding: In many cases there is a limitation as to the funding provided for building out the IBM Workplace Web Content Management infrastructure. In these cases you must construct your environment as soundly as possible within the budget. However, the budget will likely force you to reduce the infrastructure.
- Corporate standards: Your company standard might not allow for the creation of certain types of servers. For instance, it is quite common for smaller companies to omit the staging environment as it requires time and resources to perform the content review in this stage.

Important: As you look at these various architectures remember that these are only representative architectures based on practical experience in deploying IBM Workplace Web Content Management. There are many other architectural combinations that can be built based on your organization's needs. Use these architectures as a baseline from which to build your environment.

2.4.1 Basic

The Basic architecture is the smallest acceptable design for an IBM Workplace Web Content Management infrastructure.

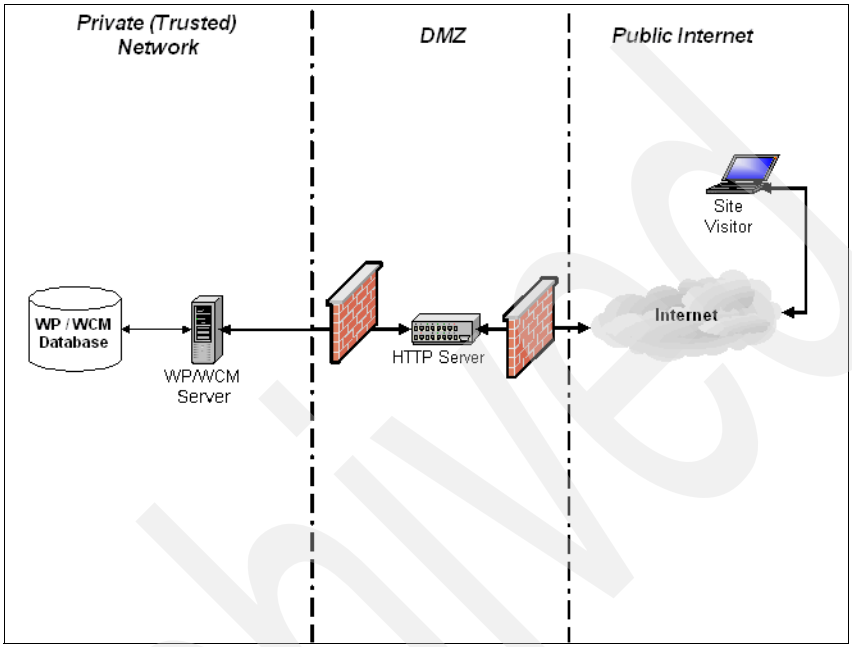


Figure 2-7 Sample architecture - Basic

In the Basic architecture, a single IBM Workplace Web Content Management server is responsible for all IBM Workplace Web Content Management activities. Site Visitors access this single server for content when they visit the Web site.

An HTTP server is placed in the DMZ to receive all requests for site content. This HTTP Server acts primarily as a proxy to shield the IBM Workplace Web Content Management server from direct access. If you use IBM HTTP Server (or Apache), the WebSphere Portal and IBM Workplace Web Content Management server can generate an HTTP Plug-in to ease configuration of the HTTP Server.

While this architecture is technically feasible, there are a variety of issues that make it sub-optimal.

Design advantages

Design advantages include simplicity. This design is very simple to implement due to the limited number of components involved.

Archived

Design disadvantages

Design disadvantages include the following:

- Failover or redundancy: This environment has no failover or redundancy in any layer. If any one part of the system were to fail the entire site would appear to be down.
- Load balancing: This environment has no load balancing capabilities. There is only one IBM Workplace Web Content Management server and it must handle the entire workload at all times.
- Maintenance: In order to perform site maintenance the site will have to be unavailable.

2.4.2 Intermediate

The Intermediate architecture attempts to correct the primary runtime deficiencies of the Basic architecture.

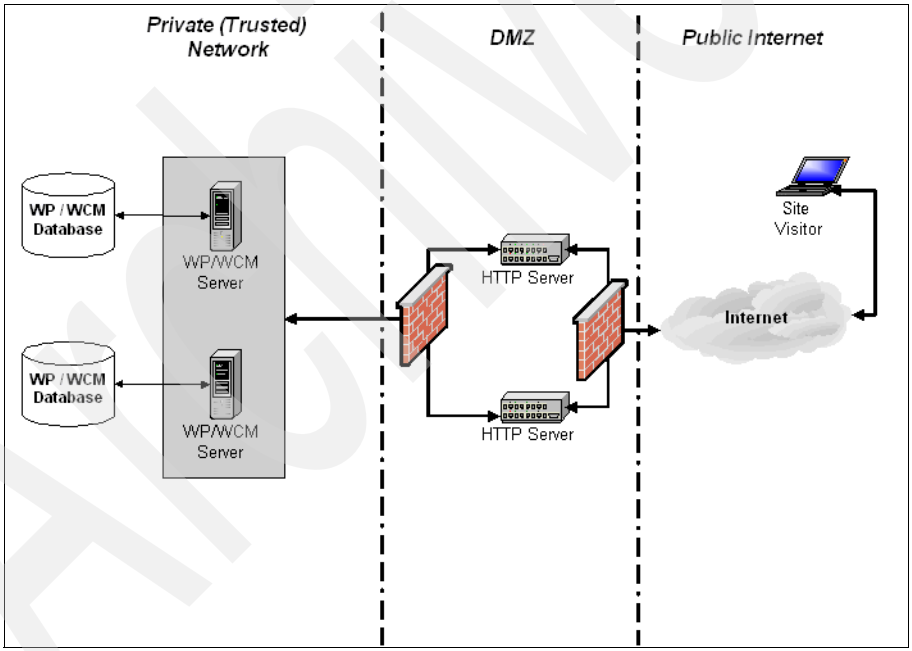


Figure 2-8 Sample architecture - Intermediate

In the Intermediate architecture, multiple IBM Workplace Web Content Management servers share the responsibility for IBM Workplace Web Content Management activities. Site Visitors access any of these servers for content when they visit the Web site.

Archived

Multiple HTTP servers are placed in the DMZ to receive requests for site content. This HTTP Servers perform two primary functions:

- Proxy: The HTTP Servers act as a proxy to shield the IBM Workplace Web Content Management servers from direct access. If you use IBM HTTP Server (or Apache), the WebSphere Portal and IBM Workplace Web Content Management server can generate an HTTP Plug-in to ease configuration of the HTTP Server.
- Load Balancing: The HTTP Plug-in can be configured to provide load balancing based on a variety of algorithms. If your IBM Workplace Web Content Management servers are not similar in processing capabilities you can set up the plug-in to load balance traffic according to server capacity.

While this architecture improves upon the Basic architecture, there are still several issues that make it sub-optimal for larger environments.

Design advantages

Design advantages include the following:

- Simplicity: This design is very simple to implement due to the limited number of components involved.
- Load balancing and redundancy: If you are using IBM HTTP Server / Apache, this environment has basic load balancing capabilities.

  Because there are duplicates of all components a basic level of redundancy exists in this design.
- Maintenance: With multiple servers available to deliver content, site maintenance can be performed without the entire site becoming unavailable.

Design disadvantages

Design disadvantages include the following:

- Administration: As more IBM Workplace Web Content Management servers are added to the infrastructure the maintenance effort to make changes increases in a linear fashion. Changes must be manually made to each server because the servers are not clustered in this design.
- Failover: While we do have redundancy with this design, there is no automated failover in the event a specific component fails. If any one part of the system were to fail there is a likelihood that the site would appear be down to some subset of the users.

Archived

2.4.3 Advanced architecture planning

The Intermediate architecture still contains one major deficiency - there is no failover within the environment. To resolve this issue we will look at building a slightly more complex infrastructure that includes **clustering** some of the servers.

WebSphere Portal and IBM Workplace Web Content Management integrated

In this architecture we continue to enable IBM Workplace Web Content Management on all of the WebSphere Portal servers. However, unlike the Intermediate architecture, in this environment we cluster the portal servers.

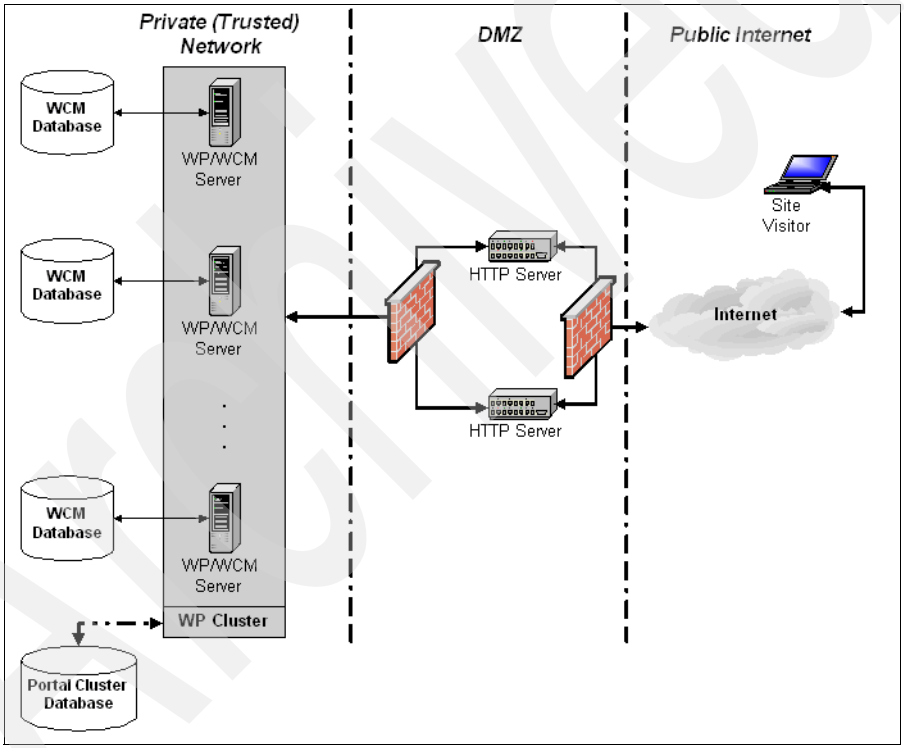


Figure 2-9 WebSphere Portal and IBM Workplace Web Content Management integrated

When you cluster WebSphere Portal servers they share identical configuration. This configuration is managed from a WebSphere Deployment Manager server (not pictured). The Deployment Manager is responsible for synchronizing the WebSphere Portal configuration across all servers in the cluster. Thus, when you deploy an updated IBM Workplace Web Content Management portlet or create a

Archived

new portal page, the Deployment Manager synchronizes all servers in the cluster — a nice reduction in administrative time and effort.

In addition, when you cluster the WebSphere Portal servers, the Deployment Manager can create a plug-in for IBM HTTP Server and Apache that allows the HTTP Servers to gracefully failover in the event that one of the portal servers stops functioning.

While setup and maintenance of a cluster does take some time and effort, this effort is typically small compared to the ongoing benefits provided to the infrastructure.

Design advantages

Design advantages include the following:

- Load balancing and redundancy: If you are using IBM HTTP Server / Apache, this environment has basic load balancing capabilities.

  Because there are duplicates of all components a basic level of redundancy exists in this design.
- Failover: The addition of a cluster allows the HTTP Server to provide failover for the WebSphere Portal servers. If any one part of the system were to fail the entire site would not appear be down.
- Maintenance: With multiple servers available to deliver content, site maintenance can be performed without the entire site becoming unavailable.
- Local rendering portlet: If IBM Workplace Web Content Management is running on all portal servers you can use the Local Rendering Portlet. By not having to communicate with an external IBM Workplace Web Content Management server (as required with the Remote Rendering Portlet) you eliminate a potential performance bottleneck.
- Administration: As more IBM Workplace Web Content Management servers are added to the infrastructure the maintenance effort to maintain the infrastructure does not increase significantly. Changes to server configurations are performed from the central Deployment Manager console.

Archived

Design disadvantages

Design disadvantages include the following:

- Complexity: This design is fairly complex. The learning curve for installing and administering a clustered environment servers should not be underestimated.
- Cost: Enabling IBM Workplace Web Content Management on every WebSphere Portal server will quickly increase your infrastructure cost. Because you get a limited entitlement of IBM Workplace Web Content Management for use in the WebSphere Portal environment, you will have to purchase additional IBM Workplace Web Content Management licenses for 5 of every 6 CPUs of WebSphere Portal.
- Authoring portlet: The authoring portlet does not function properly if run from a server participating in a cluster. Because this portlet is needed to set up Syndication, all IBM Workplace Web Content Management configuration changes controlled via the Authoring Portlet must be made before you add the IBM Workplace Web Content Management/WebSphere Portal server into the cluster. If such changes need to be made later you must remove the server from the cluster before making the changes and add it back to the cluster after the changes are complete.

Restriction: The current IBM Workplace Web Content Management software cannot take advantage of clustering with respect to the content repository. Each server that has IBM Workplace Web Content Management enabled will need to be configured to use a separate database. This is the exact opposite from the WebSphere Portal database which must be shared by all portal servers in a cluster.

Archived

WebSphere Portal and IBM Workplace Web Content Management separated

In this architecture we no longer enable IBM Workplace Web Content Management on all of the WebSphere Portal servers. Instead, we will create a separate tier of servers with the sole task of serving IBM Workplace Web Content Management content.

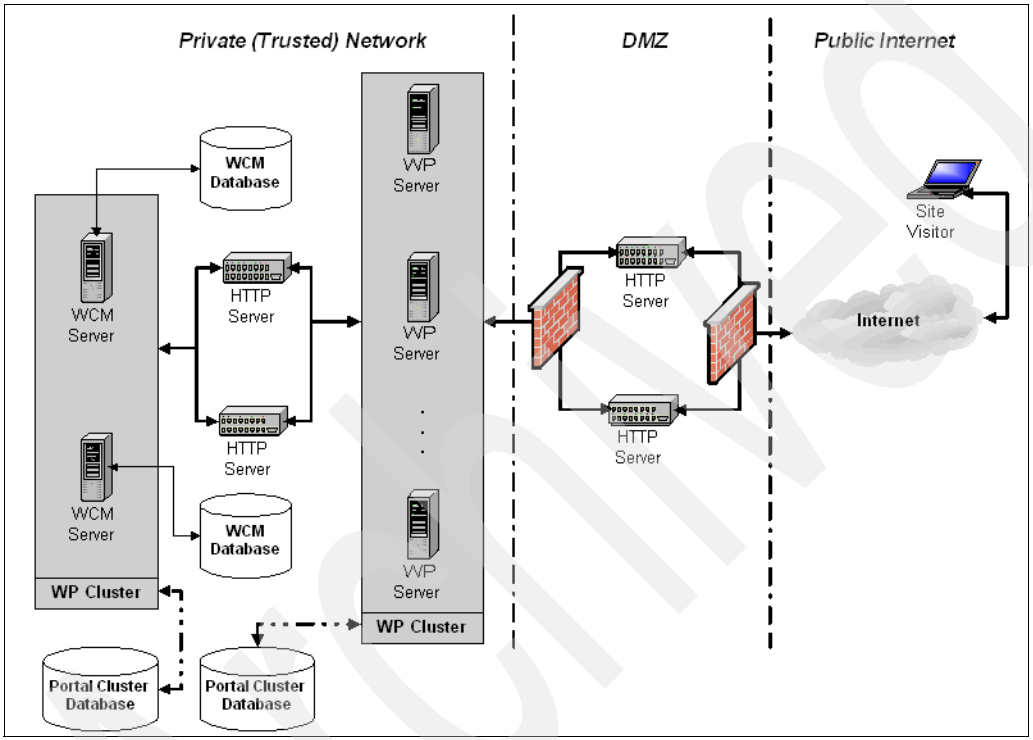


Figure 2-10 IBM Workplace Web Content Management and WebSphere Portal on separate servers

As with the other advanced architecture, this configuration clusters the portal servers. However, in this configuration we will create a separate cluster for the IBM Workplace Web Content Management servers.

Because the Authoring Portlet does not work in a clustered environment, clustering the IBM Workplace Web Content Management server presents an issue during setup and configuration. If you need to make configuration changes to a server via the Authoring Portlet you would have to remove the server from the cluster prior to making the changes - and then add the server back to the cluster. This would happen, for example, when you want to initially set up Syndication on the servers. It is recommended you complete all configuration to

all IBM Workplace Web Content Management servers before adding them to a cluster. This minimizes the issues of IBM Workplace Web Content Management and clustering.

When you cluster the WebSphere Portal or IBM Workplace Web Content Management servers they share identical configurations within their respective cluster. This configuration is managed from a WebSphere Deployment Manager server (not pictured). The Deployment Manager is responsible for synchronizing the configuration across all servers in each cluster. This means that when you deploy an updated IBM Workplace Web Content Management portlet or create a new portal page, the Deployment Manager would synchronize all servers in the appropriate cluster - a nice reduction in administrative time and effort.

In addition, when you cluster the WebSphere Portal or IBM Workplace Web Content Management servers, the Deployment Manager can create a plug-in for IBM HTTP Server or Apache that allows the HTTP Servers to gracefully failover in the event that one of the portal servers stops functioning. This is how both set of HTTP Servers can provide failover for WebSphere Portal and IBM Workplace Web Content Management requests.

Note: It is technically possible to have a single set of HTTP Servers act as the front end for all IBM Workplace Web Content Management and WebSphere Portal servers. This would require a manual modification to the plug-in that combines elements from the plug-ins created by each cluster.

While setup and maintenance of a cluster does take some time and effort, this effort is typically small compared to the ongoing benefits provided to the infrastructure.

Design advantages

Design advantages include the following:

- Cost: This design allows you to add WebSphere Portal and IBM Workplace Web Content Management processors independently - based on performance and workload requirements. As opposed to automatically adding an IBM Workplace Web Content Management processor every time you add a WebSphere Portal processor, you can add each in an independent fashion.

  Because you get a limited entitlement of IBM Workplace Web Content Management for use in the WebSphere Portal environment, this provides for better cost control as the infrastructure grows.
- Load balancing and redundancy: If you are using IBM HTTP Server or Apache, this environment has basic load balancing capabilities across each cluster.

Because there are duplicates of all components a basic level of redundancy exists in this design.

- Failover: The addition of a cluster allows the HTTP Server to provide failover for each cluster. If any one part of the system were to fail the entire site would not appear be down.
- Maintenance: With multiple servers available to deliver content, site maintenance can be performed without the entire site becoming unavailable.
- Administration: As more WebSphere Portal or IBM Workplace Web Content Management servers are added to the infrastructure the maintenance effort to maintain the infrastructure does not increase significantly. Changes to server configurations are performed from the central Deployment Manager console.

Design disadvantages

Design disadvantages include the following:

- Complexity: This design is fairly complex. The learning curve for installing and administering a clustered environment servers should not be underestimated.
- Remote Rendering Portlet: If IBM Workplace Web Content Management is running on a separate set of servers all portal servers you must use the Remote Rendering Portlet. Because this portlet communicates with an external IBM Workplace Web Content Management server there is a potential performance bottleneck if the network connection is poor.
- IBM Workplace Web Content Management Administration: The Authoring Portlet does not function properly if run from a server participating in a cluster. Because this portlet is needed to set up Syndication, all IBM Workplace Web Content Management configuration changes controlled via the Authoring Portlet must be made before you add the IBM Workplace Web Content Management server into the cluster. If such changes need to be made later you must remove the server from the cluster before making the changes and add it back to the cluster after the changes are complete.

Archived

Note: There is a cost-benefit analysis with regard to clustering stand-alone IBM Workplace Web Content Management servers. The Deployment Manager automatically creates the plug-in for the HTTP Server. This plug-in make for easy setup of failover within the IBM Workplace Web Content Management tier.

However, there are operational issue with IBM Workplace Web Content Management and clustering such as an inability to Author content or make configuration changes via the Authoring Portlet. If the clustering issues have you leaning towards not clustering the IBM Workplace Web Content Management servers there are other techniques that can provide failover for the IBM Workplace Web Content Management server tier. For example, several vendors make hardware that can perform this task. Of course this hardware has additional costs and configuration issues. There is no single answer that will work in all scenarios.

Restriction: The current IBM Workplace Web Content Management software cannot take advantage of clustering with respect to the content repository. Each server that has IBM Workplace Web Content Management enabled will need to be configured to use a separate database. This is the exact opposite from the WebSphere Portal database which must be shared by all portal servers in a cluster.

2.5 Example infrastructure

In this section, we combine the concepts that are described throughout this chapter and build an example infrastructure for the River Bend Coffee and Tea Company.

Note: Because River Bend is a fictitious company, all of the assumptions and requirements were created solely for the purpose of this discussion. In a real world situation there would be far more issues to take into consideration when designing an infrastructure.

2.5.1 Environments

Due to River Bend's small size and single office location, we implement the following environments:

- Authoring: River Bend has a single office location and a small number of content creators (~10). All content creation and management will be handled

by employees at this location. With these parameters in mind, a single Authoring server will be sufficient for the needs of River Bend.

Due to the small number of developers (2) at River Bend, this same server will also serve as the Development environment.

- Production: River Bend want to build a production infrastructure with redundancy across each element in an effort to avoid downtime. With this in mind, River Bend will build a clustered environment.

  Because the majority of all content on the site comes from IBM Workplace Web Content Management, River Bend will implement IBM Workplace Web Content Management and WebSphere Portal in an integrated fashion as described in Chapter 9, “Advanced integration with WebSphere Portal” on page 403.

- Staging: The Staging environment will be used by River Bend for review of content as well as performance testing of the overall infrastructure. With this in mind, the Staging environment must mirror all functional elements of the staging environment, including the cluster and failover implementations.

Archived

2.5.2 The River Bend infrastructure

This section reviews a sample physical infrastructure to meet the needs of The River Bend Coffee and Tea Company.

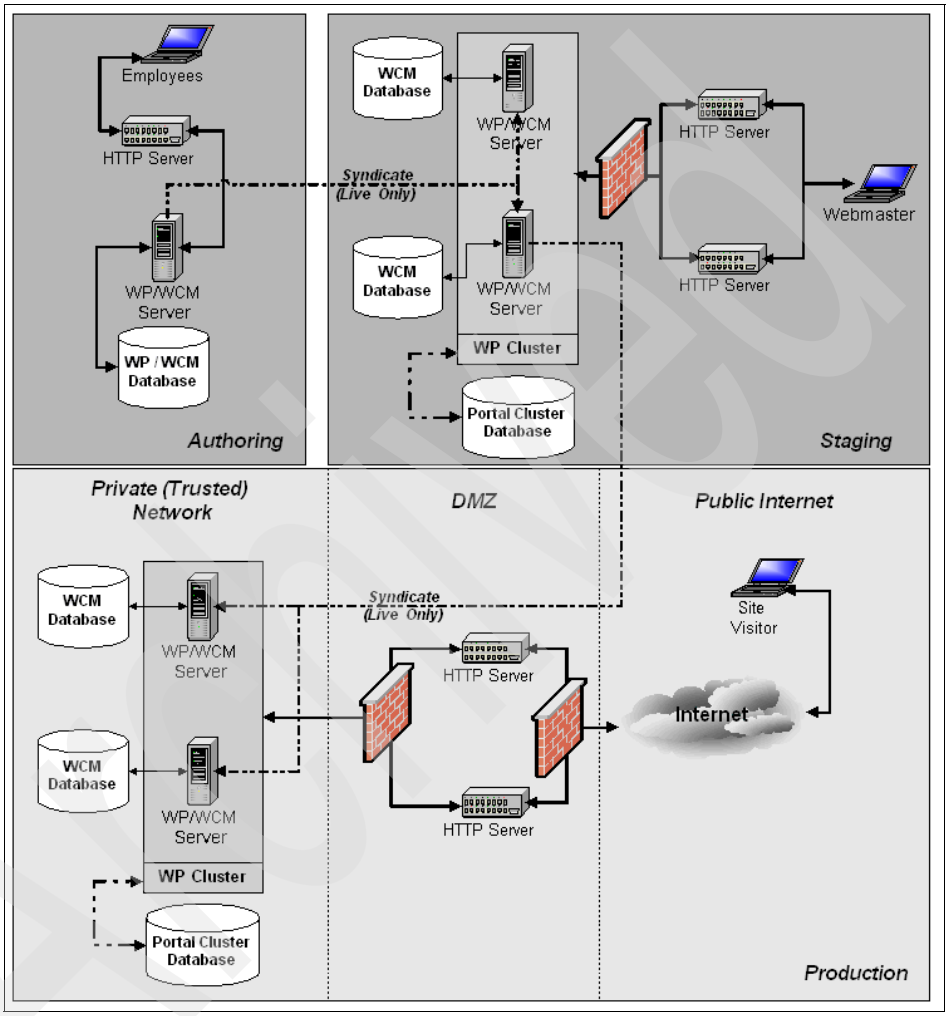


Figure 2-11 Infrastructure for River Bend Coffee and Tea Company

Authoring and development

Due to the small size of content contributors, content approvers, and developers, a single server is utilized for Authoring (and Development). An HTTP Server is placed in front of this single server.

While the HTTP Server functions could be handled by the WebSphere Portal or IBM Workplace Web Content Management server's embedded WebSphere Application Server we recommend running a separate HTTP Server.

Note: This design forces all employees through the HTTP Server as opposed to going directly to the IBM Workplace Web Content Management server. If you need to perform maintenance on the IBM Workplace Web Content Management server you want to ensure that people do not attempt to access the IBM Workplace Web Content Management server during such maintenance. The HTTP Server allows you this control. During site maintenance you can either shut down the HTTP Server (effectively making the site appear down) or place a friendly message page on the HTTP Server. In either event you will not accidentally have people accessing the IBM Workplace Web Content Management server at inappropriate times.

Production

As stated in 2.5.2, "The River Bend infrastructure" on page 65, River Bend wants to provide for a reasonable amount of failover and redundancy within their infrastructure. Furthermore, River Bend wishes to implement typical security best practices to secure their infrastructure.

Private (trusted) network

The trusted network is not directly accessible to people from outside the River Bend offices. A firewall will be used to secure the trusted network from such access.

To provide redundancy, load balancing, and failover of IBM Workplace Web Content Management content, we will implement 2 WebSphere Portal and IBM Workplace Web Content Management servers in a clustered configuration. Clustering the servers allows for the automated generation of an HTTP Plug-in file that automatically handles load balancing and failover across all WebSphere Portal and IBM Workplace Web Content Management servers in the cluster. This plug-in file is generate from the Deployment Manager server (not pictured).

Demilitarized zone (DMZ)

In order for consumers to access the River Bend Web site they will type a URL into their Web browser. From a security perspective, we do not want such traffic to directly access our trusted network. Instead, we will create a DMZ for our two HTTP Servers. The first set of firewalls will allow for direct access to the HTTP Servers from the Internet - but only on applicable ports such at 80 (HTTP) and 443 (HTTPS).

When the HTTP Servers receives a request for IBM Workplace Web Content Management content they will pass the request through the second firewall to the

IBM Workplace Web Content Management servers. The second firewall will be configured to allow traffic to the IBM Workplace Web Content Management servers only from the HTTP Servers - this will stop site visitors from directly accessing servers in the trusted network.

The HTTP Servers will have a plug-in file provided to them by the WebSphere Portal and IBM Workplace Web Content Management cluster Deployment Manager. This plug-in will automatically configure the HTTP Servers to handle requests for the WebSphere Portal and IBM Workplace Web Content Management servers. In addition, the plug-in will provide basic failover capabilities in the event one of the IBM Workplace Web Content Management servers crash.

Attention: The plug-in can be manually modified if the default settings are not appropriate, but any manual settings need to be repeated in the plug-in is later recreated.

Internet

The Internet is where site visitors will initiate requests for the River Bend site. Our DMZ must be properly configured to allow access to the HTTP servers from the Internet (and any other valid source location such as the company intranet).

Syndication

Syndication will be used to move our IBM Workplace Web Content Management content between the various environments in the infrastructure. The Syndication flow for our infrastructure is fairly straightforward.

Authoring to staging

Because we have a single Authoring server, we need to set up Syndication from this server to each Staging server. Because staging is not an appropriate location for content modification, we will only syndicate Live content to the Staging servers.

As content is approved and new technical assets are created, the Staging environment will collect syndicated changes from the Authoring environment. During normal operation, the syndication from Authoring to Staging is enabled.

Staging to production

When content is reviewed in the Staging area we need to Syndicate to our Production servers. To facilitate this we will configure one of the Staging servers to Syndicate its content out to all Production servers.

Because the Staging environment collects syndicated changes from Authoring in real time, we do not want to set up a similar Syndication pattern from Staging to Production. If we did, approved content on the Authoring server would syndicate to Staging and then to Production automatically, which defeats the purpose of having the Staging environment.

The only way to stop this from happening is to set up the Syndication from Staging to Production but then **disable** the Syndicator. When the webmaster is ready to confirm a set of changes they would perform the following process:

1. Disable Syndication from Authoring to Staging.

   When the webmaster starts review the content and design changes you do not want additional data showing up in the middle of the testing process.

2. Review content in Staging as needed.
3. Enable Syndication from Staging to Production.

   This allows the reviewed and approved changes to syndicate out to the production servers where they are viable to Site Visitors.

4. Disable Syndication from Staging to Production.

   When the content is pushed out to the Production servers you want to disable this syndication to make sure no addition content syndicates to Production accidentally.

5. Enable Syndication from Authoring to Staging.

   When the Production servers are updated we want to start aggregating the next set of changes on the Staging server.

Important: Syndication is an important process. You should plan out your Syndication strategy carefully prior to implementation.

Note: We kept the example infrastructure in this chapter moderately simple in an effort to ensure that we explained the key issues around IBM Workplace Web Content Management clearly. There are certainly far more complex infrastructures in place today for working with IBM Workplace Web Content Management. You can extrapolate the concepts that we describe in this chapter into larger designs as needed.

Archived

3

Installing and configuring IBM Workplace Web Content Management

This chapter provides an overview of the process for installing and configuring IBM Workplace Web Content Management 5.1. It includes the following sections:

- Installation overview
- Installing IBM Workplace Web Content Management
- Configuring IBM Workplace Web Content Management with DB2 Universal Database
- Applying fixes for IBM Workplace Web Content Management 5.1
- Installing the Rendering Portlets
- Post-installation configuration
- Static pre-rendering
- Removing IBM Workplace Web Content Management

© Copyright IBM Corp. 2006. All rights reserved.

3.1 Installation overview

In this chapter we will enable the IBM Workplace Web Content Management functionality on an existing WebSphere Portal 5.1 server. Installing WebSphere Portal 5.1 is a pre-requisite for enabling IBM Workplace Web Content Management 5.1. The WebSphere Portal 5.1 installation deploys all the code necessary to enable IBM Workplace Web Content Management.

There are several good resources that describe installing WebSphere Portal 5.1 including *IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook Series*, SG24-6195, which is available at:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246195.html?Open

Given this information already exists we will not repeat it in this Redbook. The following assumptions were used when building the baseline for our installation instructions:

- You have already installed WebSphere Portal 5.1
- The WebSphere Portal 5.1 server is already configured to use an LDAP
- The WebSphere Portal 5.1 server is already configured to use a remote DB2 server for the WebSphere Portal databases
- The DB2 Connect Client was used (as described in the WebSphere Portal Information Center) to connect WebSphere Portal to the remote DB2 server.

Note: When we refer to a *remote* DB2 server (or instance,) we are referring to an environment where DB2 is running on a different physical server from WebSphere Portal and IBM Workplace Web Content Management. This type of configuration is very common when you have more than one WebSphere Portal server.

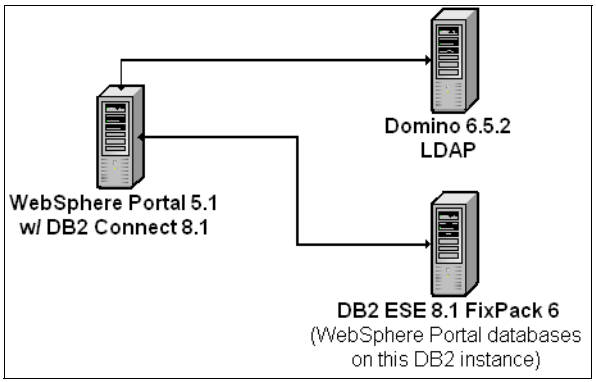


Figure 3-1 Server components used in a typical baseline installation

3.1.1 The IBM Workplace Web Content Management Server components for this book

This section provides installation instructions that are based on the following hardware and software components in the Cambridge ITSO lab:

- Operating System: All hardware uses Microsoft Windows® Advanced Server with Service Pack 4
- WebSphere Portal: The IBM Workplace Web Content Management server is a standard WebSphere Portal 5.1.0 installation. No WebSphere Portal service packs were applied.
- DB2: The remote DB2 server uses IBM DB2 Universal Database™ Enterprise Server Edition Version 8.1 with Fix Pack 6
- DB2 Connect Client: The IBM Workplace Web Content Management server uses IBM DB2 Connect Client Version 8.1

All the instructions and figures in this chapter detail the process of installing and configuring IBM Workplace Web Content Management 5.1 in a Windows 2000 Advanced Server environment. The process to perform this installation on any UNIX® variant is very similar. The steps should be the same, but some of the commands are slightly different due to operating system variances.

3.1.2 Major Installation Changes from IBM Workplace Web Content Management 1.0 and later and 2.0 and later

With the move to IBM Workplace Web Content Management 5.1, there is a significant change in how IBM Workplace Web Content Management is

Archived

distributed and installed. Before we get into the installation details lets review some of the major changes from earlier versions.

IBM Workplace Web Content Management EAR File

IBM Workplace Web Content Management 1.x and 2.0.x were stand-alone Web applications that shipped as an Enterprise Application Archive (EAR) file. In these versions there was a separate process for installing and configuring this enterprise application. The IBM Workplace Web Content Management 5.1 application code is now installed during the standard installation of WebSphere Portal 5.1. There is no separate application or installation process.

LDAP

IBM Workplace Web Content Management 1.0 and later and 2.0 and later required their own custom user repository. This repository was the sole source of authentication for people who wanted to create or manage content via the IBM Workplace Web Content Management user interface.

IBM Workplace Web Content Management 5.1 is integrated fully with WebSphere Member Manager (as is WebSphere Portal). This integration allows IBM Workplace Web Content Management to utilize your LDAP as its source for all authentication activities.

> **Note:** IBM Workplace Web Content Management 1.x and 2.0.x could use LDAP for authenticating site visitors - but not content creators and managers. This limited LDAP integration involved writing custom Java code.

Improved configuration technique

IBM Workplace Web Content Management 1.0 and later and 2.0 and later contain the following files, which are used to control the system's configuration:

- connect.cfg
- aptrixjpe.properties
- aptrixsearch.properties

A standard configuration of these versions of IBM Workplace Web Content Management required changes to at least two of these files.

In IBM Workplace Web Content Management 5.1 the majority of all *typical* configuration changes are made in the wpconfig.properties files. The same central configuration file used for configuration of WebSphere Portal.

Behind the scenes, the same IBM Workplace Web Content Management configuration files still exist in the *<WP_Root>*/wcm/config directory, but instead of direct manipulation, you use **`WPSconfig`** (the standard portal configuration

application) to push IBM Workplace Web Content Management configuration changes to the correct file(s).

> **Important:** There are some configuration changes that still require direct manipulation of the files in the <*WP_Root*>/wcm/config directory. It is worth your time to review these files in detail so you understand the role of each file.

3.2 Installing IBM Workplace Web Content Management

This section explains the installation steps for IBM Workplace Web Content Management.

3.2.1 Base server installation

> **Note:** The following instructions assume that your WebSphere Portal 5.1 server is already installed and operational using an LDAP as well as a remote DB2 server for the Portal Databases. For information about installing and configuring WebSphere Portal 5.1, see *IBM WebSphere Portal for Multiplatforms V5.1 Handbook*, SG24-6689, which is available at:
>
> http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246689.html?Open

While our ultimate goal is to have IBM Workplace Web Content Management use a remote DB2 database for its content. It is a good idea to first enable and configure IBM Workplace Web Content Management with its embedded Cloudscape database. This will confirm two important facts:

- The appropriate group (wcmadmins) exists in the LDAP
- The IBM Workplace Web Content Management portlets (Authoring and Local Rendering) were deployed and are operational

After these steps are completed, if you have issues migrating to DB2, the issue is isolated in the database configuration.

The wcmadmins group

Prior to enabling or configuring IBM Workplace Web Content Management, you must have one group setup in your user repository (LDAP). Just as the wpsadmins group contains administrators for the WebSphere Portal environment, the wcmadmins group is the administrative group for the IBM Workplace Web Content Management environment.

By default, the wcmadmins group does not include the users in the wpsadmins group. This separation allows for more fine grained control within your WebSphere Portal and IBM Workplace Web Content Management environment.

Prior to starting the process of enabling IBM Workplace Web Content Management make sure the wcmadmins group exists and at least one user account is included in this group.

Enabling IBM Workplace Web Content Management with Cloudscape

The default database for IBM Workplace Web Content Management is Cloudscape. When WebSphere Portal 5.1 is installed it creates the appropriate Cloudscape database files in the <*WP_Root*>/wcm/wcm/ilwwcm/db directory. Because the database is already in place, the initial task of configuring IBM Workplace Web Content Management involves nothing more than deploying the appropriate portlets to multiple WebSphere Portal pages.

To enable IBM Workplace Web Content Management with Cloudscape perform the following steps:

1. Stop WebSphere Portal.
2. Open a command window or command prompt.
3. Navigate to <*WP_Root*>/config directory.
4. Run the following script:

   wpsconfig configure-wcm-authoring
5. Start WebSphere Portal.

This script does the following:

- Creates the label Web Content Management. The label is given the unique name of ILWWCM.workspace.
- Creates an Authoring page. The page is given the unique name of ILWWCM.authoring.
- Creates a Content Preview page. The page is given the unique name of ILWWCM.preview.
- Deploys the Authoring Portlet to the Authoring page.
- Deploys the Local Rendering Portlet to the Content Preview page.
- Creates URL mappings for the Authoring (wcmAuthoring) and Content Preview (wcmPreview) pages. These mappings can be used to provide simplified URLs to access the two pages.

Note: If you are new to WebSphere Portal, do not worry about the URL Mappings, Unique Names, and other ancillary activities performed by this script. The most important aspect of the script is that it sets up your IBM Workplace Web Content Management environment correctly, including the Authoring Portlet.

Example 3-1 shows the end of the output from running this script. The script uses XMLAccess to create and deploy the various WebSphere Portal assets.

Example 3-1 Partial output from configure-wcm-authoring script

```
[xmlaccess] <?xml version="1.0" encoding="UTF-8"?>
[xmlaccess] <!-- IBM WebSphere Portal/5.1 build wp510_083 exported on Fri May
27 14:18:53 EDT 2005 from wcm01/9.33.85.114 -->
[xmlaccess] <!-- 1/20 [web-app wcm.author.1001 uid=wcm.author.1001] -->
[xmlaccess] <!-- 2/20 [portlet-app wcm.author.1001.1] -->
[xmlaccess] <!-- 3/20 [portlet wcm.author.1001.1.Authoring_Portlet
name=Web Content Authoring] -->
[xmlaccess] <!-- 4/20 [web-app wcm.contentviewer.1001
uid=wcm.contentviewer.1001] -->
[xmlaccess] <!-- 5/20 [portlet-app wcm.contentviewer.1001.1] -->
[xmlaccess] <!-- 6/20 [portlet wcm.contentviewer.1001.1.Content_Viewer name=Web
Content Viewer] -->
[xmlaccess] <!-- 7/20 [content-node Content.Root.My_Portal uniquename=wps.My
Portal] -->
[xmlaccess] <!-- 8/20 [content-node Content.Root.ILWWCM
uniquename=ILWWCM.workspace] -->
[xmlaccess] <!-- 9/20 [content-node Content.Root.ILWWCM_Authoring
uniquename=ILWWCM.authoring] -->
[xmlaccess] <!-- 10/20 [component _7_0G0K8FA5DC08HMHL_F0] -->
[xmlaccess] <!-- 11/20 [component _7_0G0K8FA5DC08HMHL_FP] -->
[xmlaccess] <!-- 12/20 [component _7_0G0K8FA5DC08HMHL_G0] -->
[xmlaccess] <!-- 13/20 [portletinstance _5_0G0K8FA5DC08HMHL_CH] -->
[xmlaccess] <!-- 14/20 [content-node Content.Root.ILWWCM_Preview
uniquename=ILWWCM.preview] -->
[xmlaccess] <!-- 15/20 [component _7_0G0K8FA5DC08HMHL_FU] -->
[xmlaccess] <!-- 16/20 [component _7_0G0K8FA5DC08HMHL_FV] -->
[xmlaccess] <!-- 17/20 [component _7_0G0K8FA5DC08HMHL_G1] -->
[xmlaccess] <!-- 18/20 [portletinstance _5_0G0K8FA5DC08HMHL_CI] -->
[xmlaccess] <!-- 19/20 [url-mapping-context label=wcmAuthoring] -->
[xmlaccess] <!-- 20/20 [url-mapping-context label=wcmPreview] -->
[xmlaccess] <request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
build="wp510_083" type="update" version="5.1.0.0"
xsi:noNamespaceSchemaLocation="PortalConfig_1.3.xsd">
[xmlaccess]     <status element="all" result="ok"/>
[xmlaccess] </request>

BUILD SUCCESSFUL
Total time: 3 minutes 27 seconds
```

To confirm that the script ran successfully, log in to WebSphere Portal as someone in the wcmadmins group. When you log in, you should see the Web Content Management label in the navigation. Selecting this label should bring you to the Authoring Portlet, as shown in Figure 3-2.

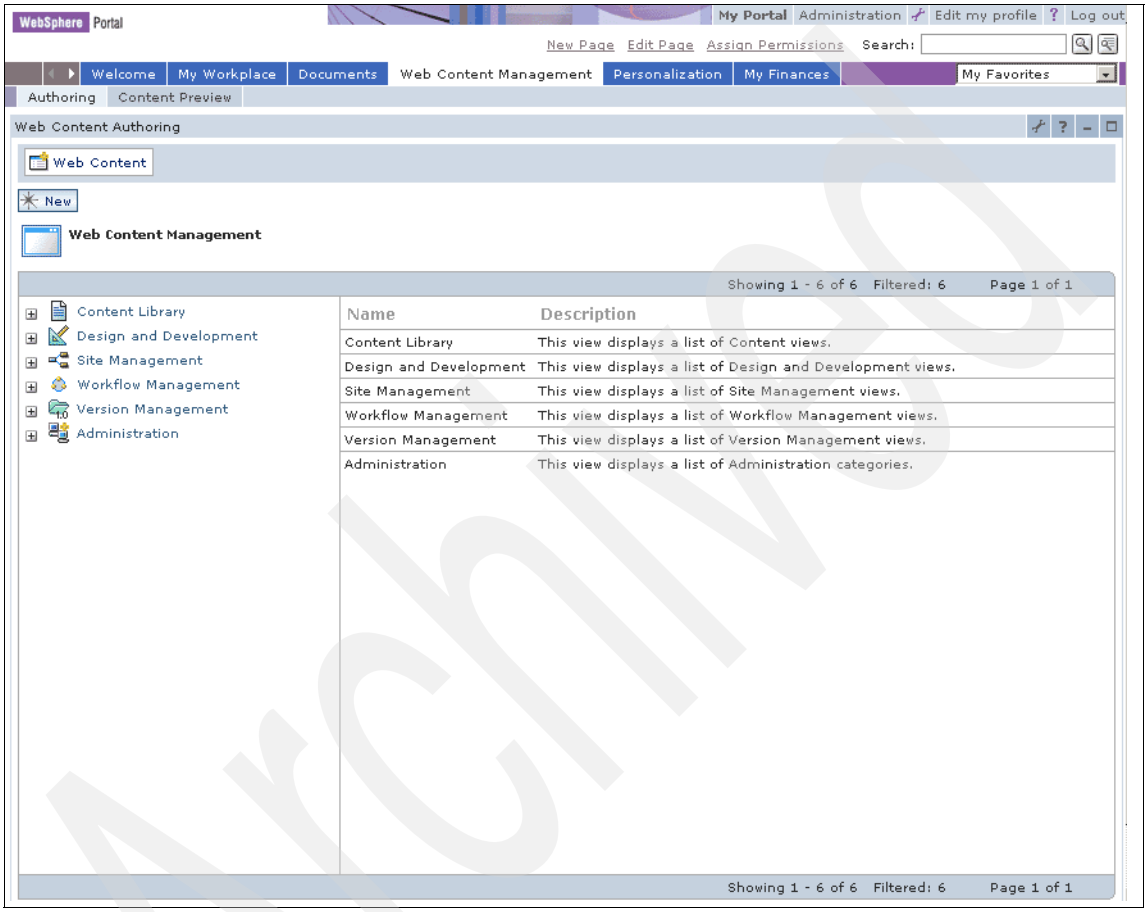


Figure 3-2 Initial access to Authoring Portlet as member of wcmadmins group

If you see a screen similar to Figure 3-2, you have successfully set up IBM Workplace Web Content Management to use the embedded Cloudscape database. At this point you are ready to move to the next step, which is - configuring IBM Workplace Web Content Management to use an external database (for example, DB2).

Attention: In some situations, you see the Authoring Portlet on the page, but you cannot see content within the portlet. If this situation occurs, confirm that you are a member of the wcmadmins group. If you added your account to wcmadmins after logging into WebSphere Portal, you need to log out and log in again before you can access to the portlet. WebSphere Portal caches your group membership when you log in. The logout and login processes force WebSphere Portal to reevaluate your group memberships.

3.3 Configuring IBM Workplace Web Content Management with DB2 Universal Database

While IBM Workplace Web Content Management provides you the embedded Cloudscape database, there are many situations where Cloudscape is inappropriate for a production environment, such as:

- Existing database standard: Your company likely already has a database platform of choice, such as DB2, Oracle, or MS SQL Server. In this situation, it makes sense to use your existing database platform as the repository for IBM Workplace Web Content Management.
- Central database servers: You plan on installing WebSphere Portal and IBM Workplace Web Content Management on multiple servers and then centralizing the databases onto a common database server. Some companies do this to simplify backup of the IBM Workplace Web Content Management data. You can certainly configure multiple instances of IBM Workplace Web Content Management to use the same database server, as long as each uses its own individual database and tables. However, Cloudscape is not *network aware*. Thus, you cannot use Cloudscape as a remote database.
- Large Web site content: If neither of these issues fit your environment, Cloudscape is certainly capable of serving up a small to medium size Web site. However, from a performance perspective, Cloudscape is not as efficient as DB2, Oracle, and so forth. If Cloudscape does not meet your performance needs, you might consider migrating your data to another database product.

This section looks at the steps that are required to change the IBM Workplace Web Content Management repository from Cloudscape to a remote DB2 server.

Archived

Note: When we refer to a *remote* DB2 server (or instance) within this book, we are referring to an environment where DB2 is running on a different physical server from WebSphere Portal and IBM Workplace Web Content Management. This type of configuration is very common when you have more than one WebSphere Portal server.

This section assumes that the remote DB2 server is already in place and our IBM Workplace Web Content Management server is already using the remote DB2 server to store the portal databases. Our process simply adds the IBM Workplace Web Content Management tables to the database server and re-configures IBM Workplace Web Content Management to use these databases. This assumption implies that the DB2 Connect Client is already installed on the server because this client is needed to migrate the WebSphere Portal databases to a remote DB2 server.

For details on migrating the core WebSphere Portal databases to a remote DB2 server, see *IBM WebSphere Portal for Multiplatforms V5.1 Handbook*, SG24-6689, which is available at:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246689.html?Open

3.3.1 Creating an IBM Workplace Web Content Management database on DB2 server

The first step in configuring IBM Workplace Web Content Management to use a remote DB2 server is to create a single database on the DB2 server. This database holds all the tables that are needed for IBM Workplace Web Content Management. To create the database, perform the following steps *on the remote DB2 server*:

1. Open a command prompt or command window.
2. At the command prompt, type **db2cmd** (on a Windows machine this opens a new command window).
3. In the new command window, enter:

   db2 "create database *db_name* using codeset UTF-8 territory US"

   where *db_name* is any valid name for a DB2 database. In this chapter, we use the name WCMDB02. For example:

   db2 "create database WCMDB02 using codeset UTF-8 territory US"

   When this command completes, you should have a new database on your DB2 server. To confirm this fact, look for the database on the DB2 server

using the DB2 Control Center (**Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**).

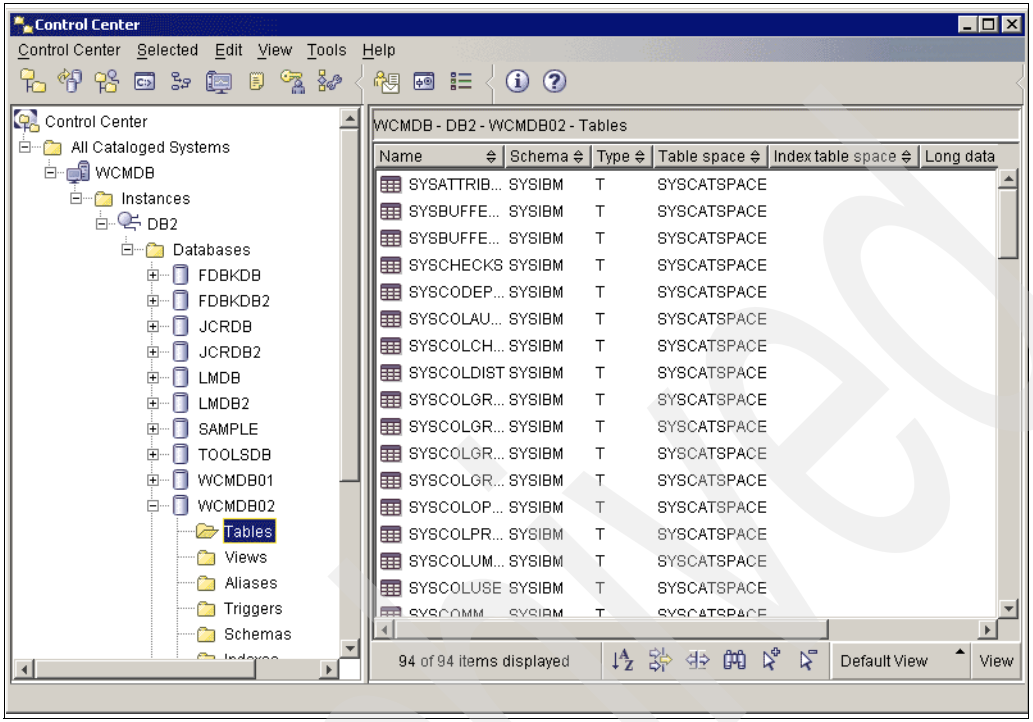


Figure 3-3 The newly created WCMDB02 database as seen in the DB2 Control Center

4. Close both command windows.

3.3.2 Configuring DB2 Connect Client on WebSphere Portal and IBM Workplace Web Content Management server

Now that we have a DB2 database in place, the next step is to make the DB2 Connect Client on the WebSphere Portal and IBM Workplace Web Content Management server aware that the database exists. This configuration is done by cataloging the database on the IBM Workplace Web Content Management server. When you catalog the database on the IBM Workplace Web Content Management server, the DB2 Connect Client creates an alias for the database. This alias is then used when connecting to the database via JDBC™. The alias makes the overall process of connecting to the database far simpler.

Creating the database alias

To create the database alias, on the WebSphere Portal and IBM Workplace Web Content Management server, follow these steps:

1. Open a command prompt or command window.
2. At the command prompt, type **db2cmd** (on a Windows machine, this command opens a new command window).
3. In the new command window, enter the following:

   db2 "catalog db *db_name* as *db_alias* at node *node_name*"

   where:

   – *db_name* is the name of the database that you created in the previous step (for example, WCMDB02).

   – *db_alias* is the local alias for the database. You can make the alias anything you want. For this example, we simply append an *A* to our database name (for example, WCMDB02A).

   – *node_name* is the alias of the remote database server. This value was created when you configured the WebSphere Portal server to use the remote DB2 server for its databases.

   In our example the full command is as follows:

   db2 "catalog db WCMDB02 as WCMDB02A at node WPDB"

Tip: If you do not know the *node_name*, you can get the value from the DB2 Configuration Assistant running on the IBM Workplace Web Content Management server (**Start** → **Programs** → **IBM DB2** → **Set-up Tools** → **Configuration Assistant**). When the Configuration Assistant starts, you see a list of all catalogued databases. Within the row of any of the core portal databases, the value in the Instance Node column is the value you need (Figure 3-4 on page 81).

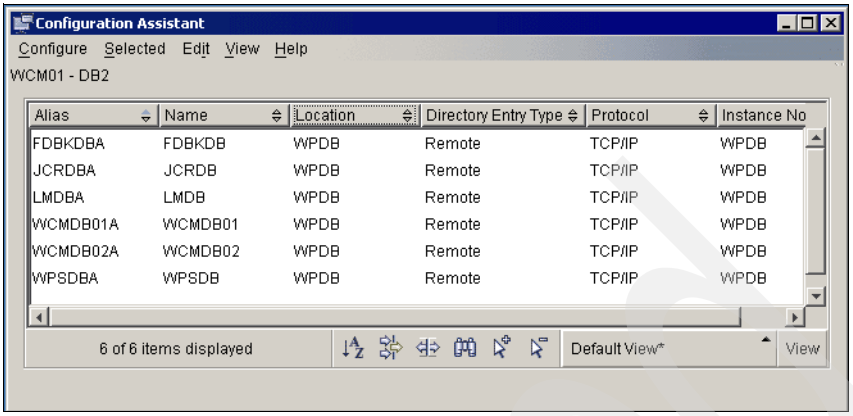


Figure 3-4 DB2 Configuration Assistant, illustrating a list of all catalogued databases

4. Close both command windows.

Testing the database catalog

After you catalog the database, you can attempt to connect to the database on the DB2 server. If you can connect to the database you have properly cataloged the database. To test the connection:

1. Open a command prompt or command window.
2. At the command prompt, type **db2cmd** (on a Windows machine this will open a new command window).
3. In the new command window, enter the following:

   db2 "connect to *db_alias* user *db_user_name* using *db_user_pwd*"

   where:

   – *db_alias* is the local alias for the database. You can make the alias anything you want. For this example, we simply append an *A* to our database name (for example, WCMDB02A).

   – *db_user_name* is a valid database user with access to the IBM Workplace Web Content Management database

   – *db_user_pwd* is the password for *db_user_name*. For our example, the command is as follows:

   db2 "connect to WCMDB02A user db2admin using db2admin"

Archived

If the connection is set up correctly, you see a screen similar to that shown in Figure 3-5.

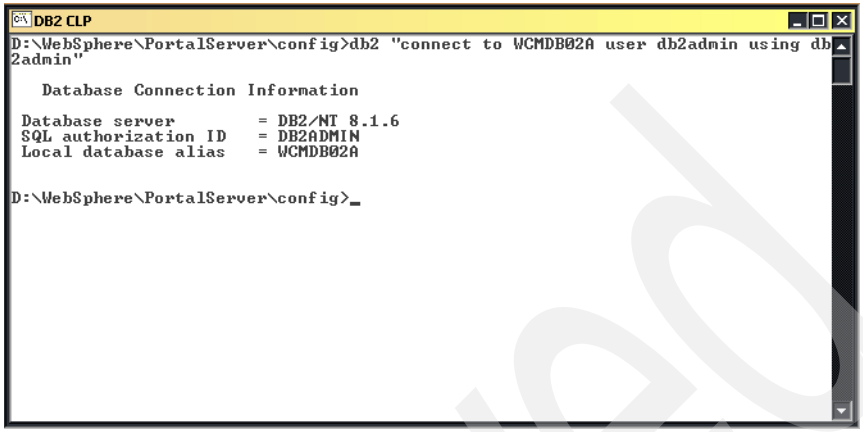


Figure 3-5 Connecting to the remote database via its alias

4. Type **`db2 "quit"`** to close the database connection.
5. Close both command windows.

Now that you have properly cataloged the remote database, it is time to configure IBM Workplace Web Content Management to use this new database.

3.3.3 Changing the configuration

The process of changing the IBM Workplace Web Content Management configuration to use the new DB2 database involves two steps.

- Updating the wpconfig.properties file with specific information about the DB2 environment.
- Running one **`wpsconfig`** task to push the values from wpconfig.properties into the IBM Workplace Web Content Management configuration files.

> **Note:** If you are new to WebSphere Portal and IBM Workplace Web Content Management, refer to the WebSphere Portal and IBM Workplace Web Content Management Information Center for an overview of the wpconfig.properties file and the **`wpsconfig`** tasks that are associated with IBM Workplace Web Content Management.

Archived

Modifying wpconfig.properties

The wpconfig.properties file contains many settings for various parts of WebSphere Portal, not just IBM Workplace Web Content Management. Before you make changes to this file, make a copy! Saving a copy can save time and trouble if you make a mistake as you edit the file.

The wpconfig.properties file contains a series of settings that need to be adjusted in order to allow IBM Workplace Web Content Management to utilize any database other than the embedded Cloudscape. There is an entire section of this file dedicated to the IBM Workplace Web Content Management settings. Many of these settings never need be changed, but in order to use our remote DB2 database we will need to change seven settings. The values we used for these seven settings on our test server are as shown in Example 3-2.

Attention: To make Example 3-2 easier to read, it was heavily abbreviated for this example. We omitted those values that you do not need to change and the examples for each setting that exists in the real properties file. The real wpconfig.properties file provides examples as comments throughout the file, such as the following:

```
# WcmDbType: The type of database to be used for Workplace Web Content
Management
# DB2: { db2 }
# Oracle:                  { oracle }
# MS SQL Server:           { sqlserver }
# Cloudscape:              { cloudscape }
# DB2 for iSeries:         { db2_iseries }
# DB2 for z/OS and OS/390: { db2_zos }
WcmDbType=db2
```

Example 3-2 Required changes for remote DB2 to the wpconfig.properties file

```
#####################################################################
# IBM Lotus Workplace Web Content Management (WCM) Properties - BEGIN
#####################################################################
# WcmDbType: The type of database to be used for WCM
WcmDbType=db2

# WcmDbName: The name of the WCM database.
WcmDbName=WCMDB02A

# WcmDbUser: The WCM database administrator user ID
WcmDbUser=db2admin

# WcmDbPassword: The WCM database administrator password
WcmDbPassword=db2admin
```

```
# WcmDbUrl: The WCM database URL
WcmDbUrl=jdbc:db2:WCMDB02A

# WcmDbDriver: The name of class SqlProcessor will use to import SQL files,
also known as "JDBC provider"
WcmDbDriver=COM.ibm.db2.jdbc.app.DB2Driver

# WcmDbLibrary: The directory and name of the zip file containing db.driver
class
WcmDbLibrary=D:/IBM/SQLLIB/java/db2java.zip
#####################################################################
# WCM Properties - END
#####################################################################
```

While most of these values are self evident, the following sections look at some of these values in detail.

WcmDbName

When we cataloged the remote IBM Workplace Web Content Management database with the DB2 Connect™ Client, we gave the database an alias. On the IBM Workplace Web Content Management server, this alias is how IBM Workplace Web Content Management identifies the database. This value must match the alias value that you created for the database. For our example database, the value becomes `WCMDB02A`.

WcmDbUser and WcmDbPassword

When you created the IBM Workplace Web Content Management databases on the DB2 server, you could (optionally) create a specific user that has access to this database. These fields need a valid DB2 user name and password that has full access to the IBM Workplace Web Content Management database. In our example, we use the user name `db2admin`, which is the default administrator for the entire DB2 installation. It is not recommended that you use this default user name in a production environment.

WcmDbUrl

The wpconfig.properties file provides examples of valid URLs for various types of databases, including DB2. The sample for DB2 is:

```
# db2:          { jdbc:db2:WCMDB }
```

The reason for this very simple URL is the DB2 Connect Client. The only portion of this URL that needs to be changed is the very right-most portion (the part in all capital letters). This portion must match the alias that you created for the database. For our example database, the URL becomes:

```
jdbc:db2:WCMDB02A
```

WcmDbLibrary

In order to use DB2 Connect and JDBC to connect to the remote database, we must make the appropriate JDBC drivers available to IBM Workplace Web Content Management. The `WcmDbLibrary` value points to the location of the db2java.zip file that was installed with DB2 Connect Client.

> **Important:** The path value uses UNIX style slashes in the directory structure, even if you are installing on Windows.

Running wpsconfig script

After we update the wpconfig.properties file, we must apply these changes to the IBM Workplace Web Content Management application.

> **Note:** If you are new to WebSphere Portal, note that the wpconfig.properties file is not used at runtime by WebSphere Portal or IBM Workplace Web Content Management. The values in this file are applied to the appropriate locations when you run one of many **`wpsconfig`** tasks. The WebSphere Portal Information Center covers the various tasks in detail.

We just enabled IBM Workplace Web Content Management, and there is no data in the Cloudscape database. So, the easiest way to switch to DB2 is to remove the current repository (Cloudscape) and then to configure IBM Workplace Web Content Management to use DB2 using the following steps:

1. Stop WebSphere Portal.
2. On the WebSphere Portal and IBM Workplace Web Content Management server open a command window or command prompt.
3. Change to the <*WP_Root*>/config directory (for example, D:\WebSphere\PortalServer\config).
4. Type `WPSconfig remove-wcm-repository`.

   At the conclusion of the task you should receive a `BUILD SUCCESSFUL` message.
5. Type `WPSconfig config-wcm-repository`.

   At the conclusion of the task you should receive a `BUILD SUCCESSFUL` message.
6. Change to the <*WP_Root*>/wcm/ilwwcm/system directory (for example, D:\WebSphere\PortalServer\wcm\ilwwcm\system), and delete all files or folders in the directory.
7. Restart WebSphere Portal.

After WebSphere Portal starts, your IBM Workplace Web Content Management environment should be operational and should use the remote DB2 server for content storage.

3.3.4 Confirming your new IBM Workplace Web Content Management configuration

This section explains how to confirm that the new IBM Workplace Web Content Management configuration was successful.

DB2 server

Upon a successful start with your new settings, IBM Workplace Web Content Management should create the two tables that it needs within the IBM Workplace Web Content Management database. You can confirm the existence of these tables by looking at the database with the DB2 Control Center on the remote DB2 server (**Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**). If the configuration was done correctly, when you look in the Control Center you should see tables named AJPE and AJPE_RESOURCES, as shown in Figure 3-6.

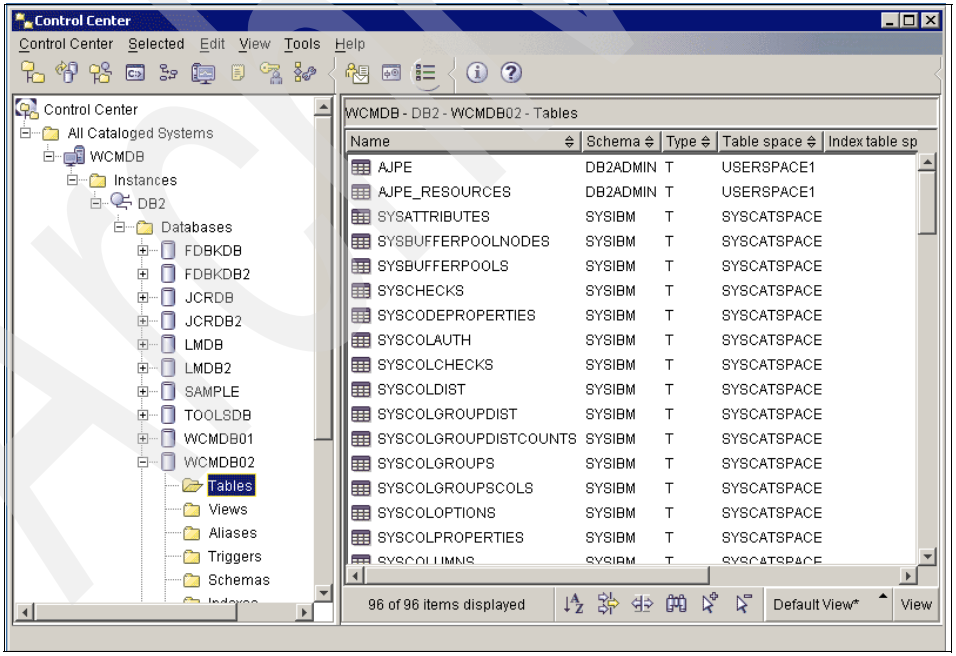


Figure 3-6 Confirming creation of IBM Workplace Web Content Management tables

WebSphere Portal logs

If the two IBM Workplace Web Content Management tables are not in the IBM Workplace Web Content Management database, you can look at the WebSphere Portal logs in <*WP_Root*>/logs. In this directory look closely at SystemOut.log, SystemErr.log, and wps_<timestamp>.log. Any of these files might contain clues as to the root of the issue.

IBM Workplace Web Content Management logs

The final place that might contain some clues regarding a failed migration to DB2 are the IBM Workplace Web Content Management logs. IBM Workplace Web Content Management maintains a set of three log files that are separate from the main WebSphere Portal logs. You can find these files — connect.log, debug.log, and error.log — in <*WP_Root*>/wcm/connect/log.

3.4 Applying fixes for IBM Workplace Web Content Management 5.1

At the time of this writing, there are several Technotes and fixes that need to be applied to every IBM Workplace Web Content Management 5.1 installation. This section discusses the installation of the necessary fixes, the manual modification that needs to be done to the IBM Workplace Web Content Management configuration file, as well as one modification to WebSphere Application Server that is required per a Technote.

3.4.1 Technote, reference #1201178

This technote only effects the IBM Workplace Web Content Management server after you enable security, which is done typically when you configure WebSphere Portal and IBM Workplace Web Content Management to use an LDAP. When this configuration change takes place, IBM Workplace Web Content Management is no longer able to search for users or groups when assigning access to content and to components. The following error message might appear in the logs:

```
"ldaphost:389/userID;accessID:
user:ldaphost:389/CN=userID,OU=Org Unit,O=Organization
is not granted any of the required roles: Everyone"
```

This error occurs if the user name under which you are trying to search does not have the correct access in WebSphere Member Manager. To correct this issue, modify the security settings for the wmmApp, one of the Enterprise Applications that is installed within WebSphere Application Server.

To update the settings, use the following steps:

1. Start server1 (the default server on every WebSphere Application Server installation).
2. Open the WebSphere Application Server Administrative console:

   `http://<servername>:9090/admin`

3. Log in to the Administrative Console.

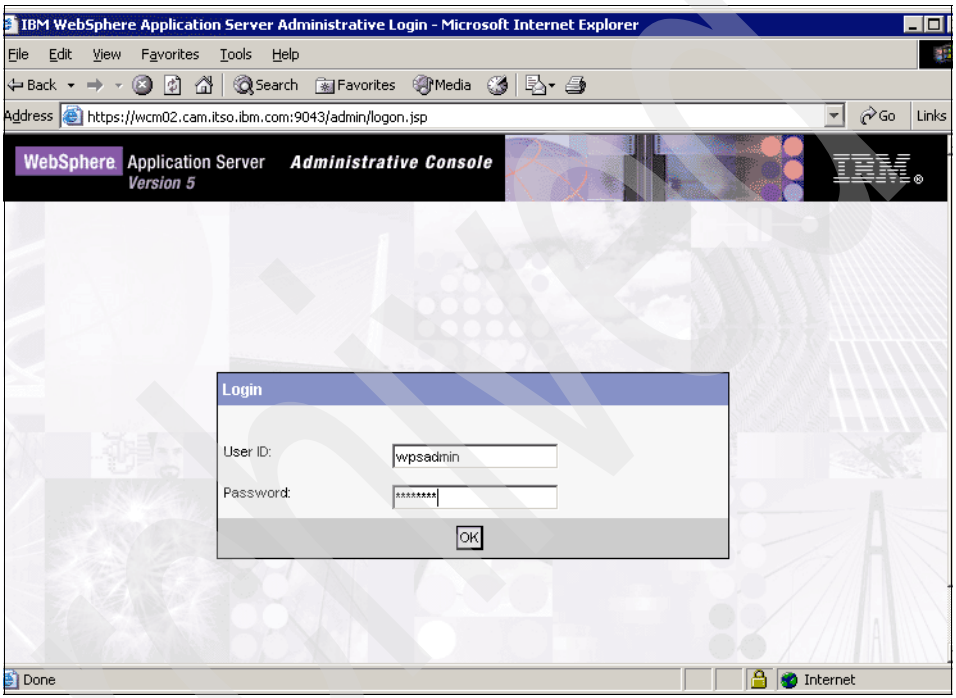


Figure 3-7 Login screen for WebSphere Application Server Console

4. Expand **Applications**, select **Enterprise Application**, and search for wmmApp. To do this quickly, use the Filter section to search for wmmApp.

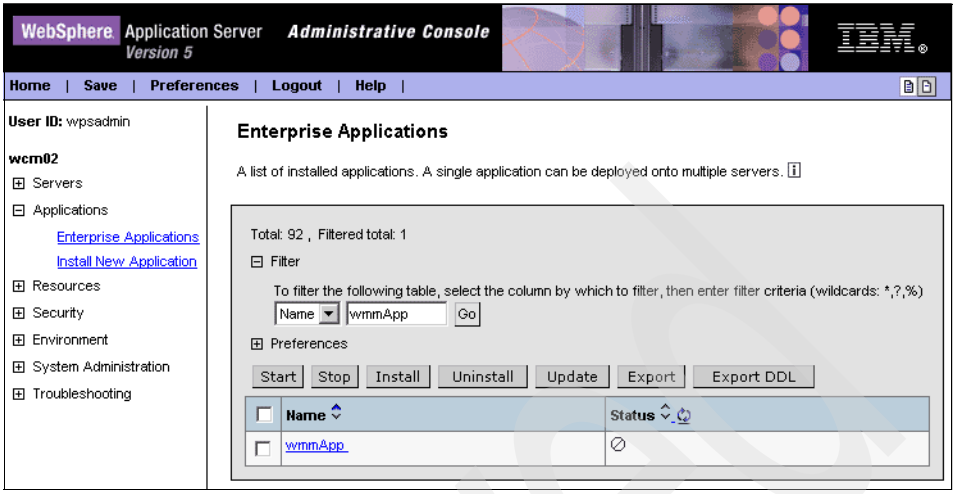


Figure 3-8 Locating the wmmApp with a filter

5. Click **wmmApp**.
6. From the Additional Properties section, click **Map Security Roles** to users or groups.

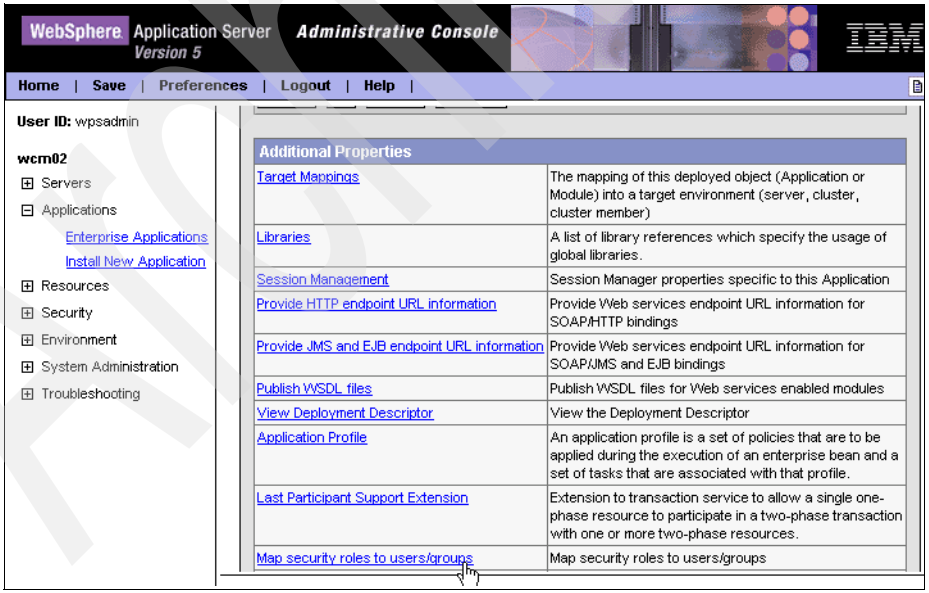


Figure 3-9 Map Security Roles option in Additional properties

Archived

7. Select the options to grant All Authenticated users access to the wmmApp Enterprise Application, as shown in Figure 3-10.

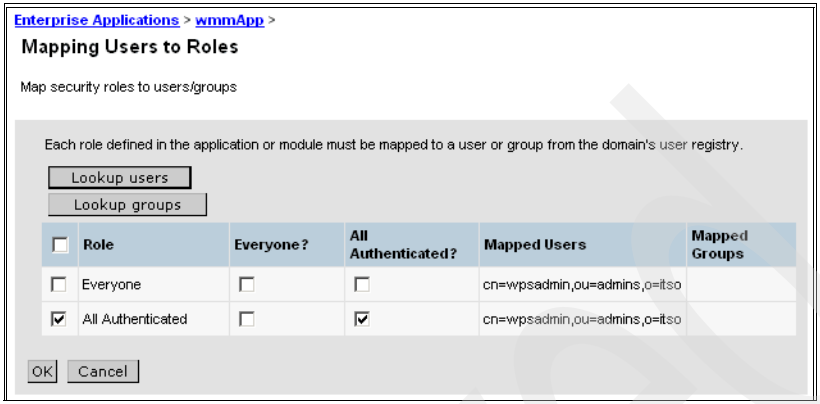


Figure 3-10 Giving All Authenticated users access to wmmApp

8. Click **OK** and then click **Save** to apply the changes to the master configuration

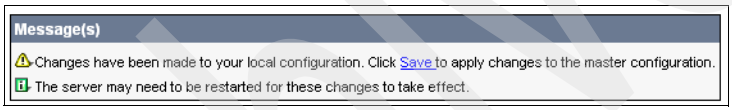


Figure 3-11 Save changes to Master Configuration

9. Click **Save** in the Save to Master Configuration Window.

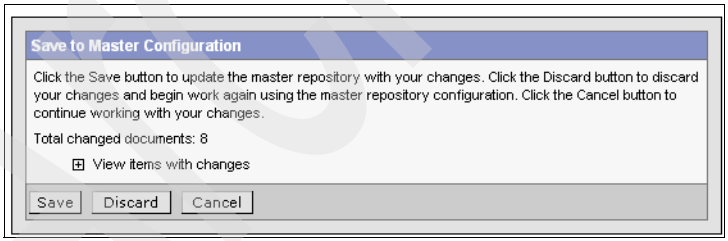


Figure 3-12 Save Master Configuration section

10. Stop and start WebSphere Portal.

You should now be able to search for users from within the Authoring Portlet.

Note: As an alternative to giving All Authenticated Users access you could create a group with all Workplace Web Content Management users and grant that group access to the wmmApp Enterprise Application.

3.4.2 IBM Workplace Web Content Management connect.cfg file

When you initially configure your WebSphere Portal and IBM Workplace Web Content Management server to use an external user repository (for example, LDAP) there is a **wpsconfig** task used to implement the security changes:

```
WPSconfig enable-security-ldap
```

This task makes numerous changes within WebSphere Portal / Workplace Web Content Management to properly configure the environment for your LDAP. Unfortunately, in WebSphere Portal 5.1 this script does not make several necessary modifications to the Workplace Web Content Management connect.cfg file. This file is located in <*WP_Root*>/wcm/config.

The missed changes occur in the `<UserManagement>` and `<JCRConnector>` portions of the file. Example 3-3 shows the fields in question and the appropriate changes for the Domino LDAP server in the ITSO lab.

Example 3-3 Adjustment for LDAP settings

```
<UserManagement>
   <AdminGroupCommonName value="wcmadmins" />
<!--
    UserPrefix: The LDAP user prefix appropriate for your LDAP server
    IBM Directory Server: { uid }
    Domino:               { cn }
    Active Directory:     { cn )
    SunOne:               { uid }
    Novell eDirectory     { uid }
-->
    <UserPrefix value="cn" />
<!--
    GroupPrefix: The LDAP group prefix appropriate for your LDAP server
    IBM Directory Server: { uid }
    Domino:               { cn }
    Active Directory:     { cn )
    SunOne:               { uid }
    Novell eDirectory     { uid }
-->
    <GroupPrefix value="cn" />
    <LoginPageLocation
value="http://wcm02.cam.itso.ibm.com:9081/wps/wcm/webinterface/login/login.jsp"
/>
    <UserCommonNameAttribute value="cn" />
</UserManagement>
.
.
.
<JCRConnector>
```

Archived

```
      <DefaultRepository value="ibmjcr/Repository"/>
      <DefaultUsername   value="cn=wpsadmin,ou=admins,o=itso"/>
      <DefaultPassword   value="passw0rd"/>
      <DefaultWorkspace  value=""/>
      <ConnectionTimeout value = 300 />
      <ConnectionManager>
         <MaxConnectionPools value=1000 />
         <MaxConnectionsPerPool value=64 />
      </ConnectionManager>
</JCRConnector>
```

After you make these changes, you must restart the portal server for the changes to take effect.

3.4.3 IBM Workplace Web Content Management fix PK02213

The other change you should make to any standard IBM Workplace Web Content Management 5.1 installation is contained in fix PK02213. According to the `ReeadMe.txt` file for the fix, the problem definition is:

> LDAP user can log into Portal but can't see any of the views in the Workplace Web Content Management authoring portlet. Workplace Web Content Management is configured to use an LDAP repository. An LDAP user logs into portal successfully but when they navigate to the Workplace Web Content Management Authoring Portlet page they cannot see the views in the portlet.“

Clearly, you need to address this issue. However, you should note that this fix has three pre-requisite fixes that you must also install, in the following order:

1. PQ98531
2. PK00404
3. LO06953
4. PK02213

Important: Before you install these fixes, make sure that you download the latest Portal Update Installer. If you have an older version of the installer, some of these fixes will not install. You can get the latest installer from:

http://www-1.ibm.com/support/docview.wss?rs=688&context=SSHRKX&q1=portal+update+installer&uid=swg24006942&loc=en_US&cs=utf-8&lang=en

If this does not work for you, go to the WebSphere Portal Support Page at the following Web site and search for *portal update installer*:

http://www-306.ibm.com/software/genservers/portal/support/

These installation steps assume that you have already downloaded the Portal Update Installer and have set it up as described in the portal_update_installer.html file that is included with the Portal Update Installer.

Very Important: There are small nuances that are associated with running the Portal Update Installer on various operating systems. Be sure that you read portal_update_installer.html prior to starting the install process!

Installing the PQ98531 IBM Workplace Web Content Management fix and its prerequisites

Each of these fixes for IBM Workplace Web Content Management follows the same format. IBM provides the fix as a zipped file. The zipped file contains the fix itself as a .jar file as well as a ReadMe.txt file that explains the purpose for the fix.

Important: Normally, you would want to install each fix individually (including performing the post-install steps that are described in the ReadMe.txt file for each fix) prior to installing the next fix. In the case of these fixes, the first fix has no post-installation tasks and the remaining three fixes have exactly the same post-installation tasks. Thus, you can install all of these fixes in the proper sequence and then do the post-installation tasks at one time.

If you are not sure that this type of situation exists with other fixes, install each one individually and perform the remaining post-installation steps before installing the next fix.

To install this group of fixes:

1. Open a command window or command prompt.
2. Change to the <*WAS_root*>/bin directory, where <*WAS_root*> is the installation directory of the WebSphere Application Server that is associated with WebSphere Portal (for example, D:\WebSphere\AppServer\bin).
3. Type `setupCmdLine`.

   This command sets up a variety of environment variables to ensure the Portal Update Installer uses the correct WebSphere and Java environment. *Make sure that you run all of the remaining commands in this section in this same command window*. These environment variables must be set correctly for the Portal Update Installer to work properly.
4. Change to the <*WP_Root*>/update directory.

5. Enter the following:

```
updatePortal -fix -installdir "D:\WebSphere\PortalServer" -fixDir
"D:\WebSphere\PortalServer\update\fixes" -install -fixes PQ98531
```

   Replace the *installdir* and *fixDir* paths with the ones that applicable for your environment.

6. Review the output to confirm that there were no errors.
7. Enter the following:

```
updatePortal -fix -installdir "D:\WebSphere\PortalServer" -fixDir
"D:\WebSphere\PortalServer\update\fixes" -install -fixes PK00404
```

   Replace the *installdir* and *fixDir* paths with the ones that are applicable for your environment.

8. Review the output to confirm that there were no errors.
9. Enter the following:

```
updatePortal -fix -installdir "D:\WebSphere\PortalServer" -fixDir
"D:\WebSphere\PortalServer\update\fixes" -install -fixes LO06953
```

   Replace the *installdir* and *fixDir* paths with the ones that are applicable for your environment.

10. Review the output to confirm that there were no errors.
11. Enter the following:

```
updatePortal -fix -installdir "D:\WebSphere\PortalServer" -fixDir
"D:\WebSphere\PortalServer\update\fixes" -install -fixes PK02213
```

   Replace the *installdir* and *fixDir* paths with the ones that are applicable for your environment.

12. Review the output to confirm that there were no errors.

Now that these fixes are installed, you need to perform the post-installation steps (which are identical for the last three fixes) as follows:

1. Delete the Workplace Web Content Management indexes by deleting all contents from the <*WP_Root*>/wcm/ilwwcm/system directory.
2. Start WebSphere Portal.
3. Change to the <*WP_Root*>/config directory.
4. Type `WPSconfig remove-wcm-authoring`.
5. Type `WPSconfig configure-wcm-authoring`.
6. (*optional*) If you have deployed other copies of the Authoring or Rendering Portlets on the IBM Workplace Web Content Management server, re-deploy them manually through the WebSphere Portal Administration interface.

Archived

As the portal server restarts, it re-creates the IBM Workplace Web Content Management indexes because they are no longer in the <*WP_Root*>/wcm/ilwwcm/system directory. If you install a fix on an IBM Workplace Web Content Management server that contains a large amount of IBM Workplace Web Content Management content, this process can take some time. During this time, the portal is not available for users to log in because the process is part of the Portal startup process. Be aware of this issue when you install IBM Workplace Web Content Management fixes, because the restart time is typically much longer than the time to install the fix itself.

3.5 Installing the Rendering Portlets

When your IBM Workplace Web Content Management server is operational, you will likely want to install at least one instance of an IBM Workplace Web Content Management Rendering Portlet so that WebSphere Portal site visitors can see content. To help ease the process of showing IBM Workplace Web Content Management content in WebSphere Portal, IBM Workplace Web Content Management ships with the following rendering portlets:

- Local Rendering Portlet: Use this portlet when IBM Workplace Web Content Management is running on the same server as the instance of WebSphere Portal that will show the content.
- Remote Rendering Portlet: Use this portlet when IBM Workplace Web Content Management is running on a different server than the instance of WebSphere Portal that will show the content.

3.5.1 Local Rendering Portlet

If you configure IBM Workplace Web Content Management on a WebSphere Portal server, the Local Rendering portlet can be found at:

```
<WP_Root>/wcm/installableApps
```

The Local Rendering Portlet is in a Web archive file named ilwwcm-localrendering-portlet.war.

Deploying the Local Rendering Portlet

Because using the Local Rendering Portlet only makes sense on a server where IBM Workplace Web Content Management is enabled, we can make one assumption for this portlet — *it is already installed on the IBM Workplace Web Content Management server*. This assumption is true because the Content Preview page that was created when we first enabled IBN Workplace Web

Content Management (WPSconfig configure-wcm-authoring) uses the Local Rendering Portlet.

Because the portlet is already installed on the server, we cannot install it again. If we try to install it, the WebSphere Portal administrative console fails with an error similar to that shown in Figure 3-13.

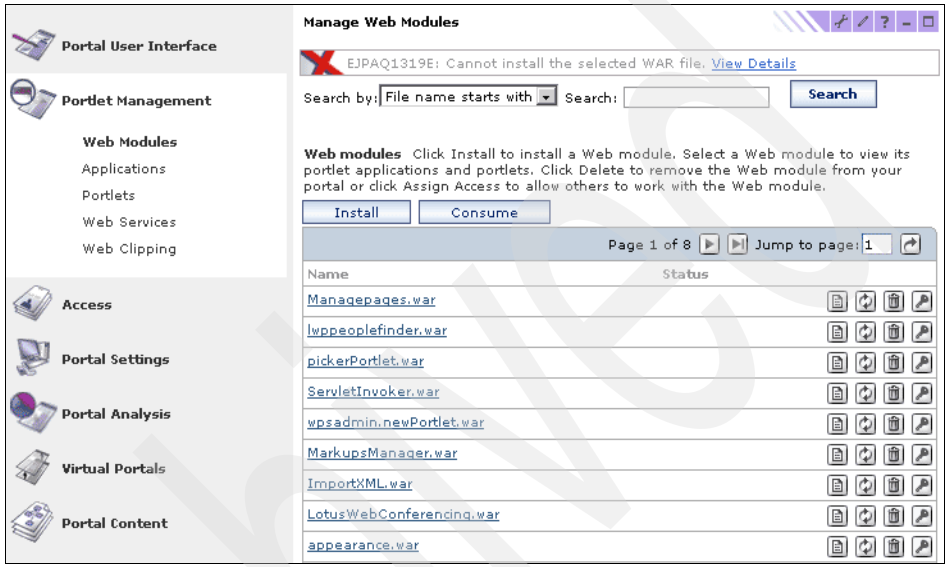


Figure 3-13 Error message when trying to install an already installed portlet

If you click **View Details**, the first part of the message detail provides information that is similar to the following message:

```
EJPAQ1319E: Cannot install the selected WAR file.

com.ibm.wps.command.applications.AppAlreadyExistsException: EJPPF0170E:
Error occurred. In the XML element portlet-app Web Content Management -
Content Viewer, the uid=wcm.contentviewer.1001 already exists for stored
abstract portlet application WCM Local Rendering Portlet.
```

Instead of trying to reinstall the portlet, all we need to do is make a copy of the portlet so we can deploy it on another page.

Copying the Local Rendering Portlet

To copy the existing Local Rendering Portlet:

1. Log in to WebSphere Portal as an Administrator (for example, a member of wpsadmins group).

2. Click the Administration link to access the Administration Console within WebSphere Portal. This link is typically found in the upper right corner of the page if you are using one of the standard WebSphere Portal themes.
3. In the left hand navigation, click **Portlet Management** → **Portlets** to access the Manage Portlets GUI, as shown in Figure 3-14.

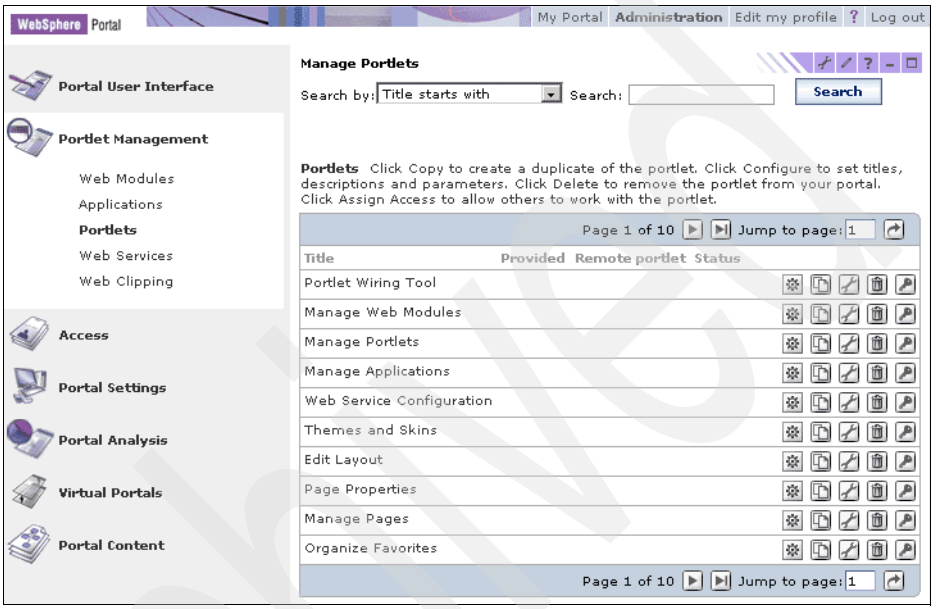


Figure 3-14 Manage Portlets GUI

4. Before you can copy the Local Rendering Portlet you must first locate the portlet. The easiest way to do that is to search for the portlet. In the Manage Portlets GUI select Title contains in the Search by select list and type `Web Content` in the Search text box. This action returns a short list including the Web Content Viewer , which is the Title of the Local Rendering Portlet, as shown in Figure 3-15 on page 98.

Archived

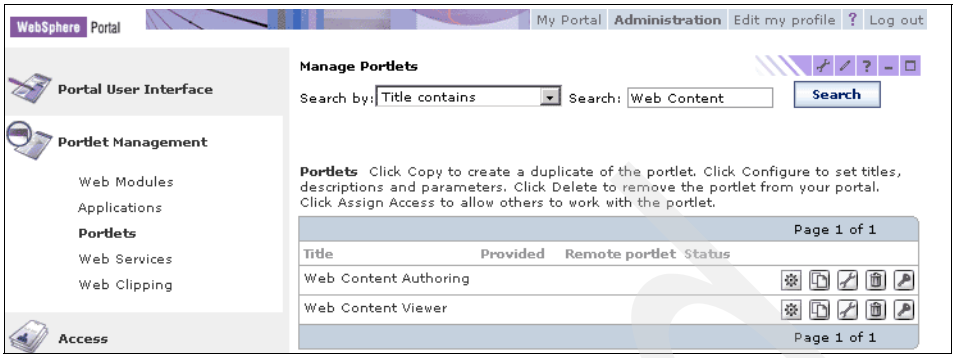


Figure 3-15 Results of searching for Web Content

5. Make a copy of the Web Content Viewer portlet by clicking the Copy icon, as shown in Figure 3-16.

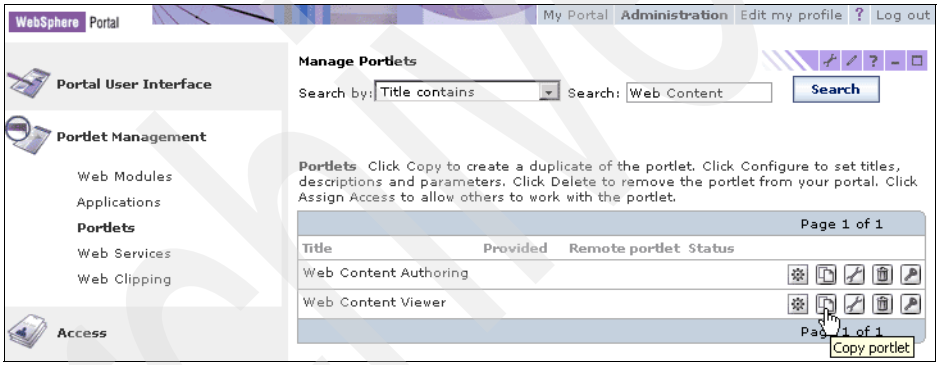


Figure 3-16 Copying the Web Content Viewer portlet

Archived

6. You are prompted to provide a new name for the Portlet Application and Portlet that is associated with the copy you are creating (Figure 3-17). The system provides default values that are based on the values in the existing portlet. Replace these defaults with something meaningful.

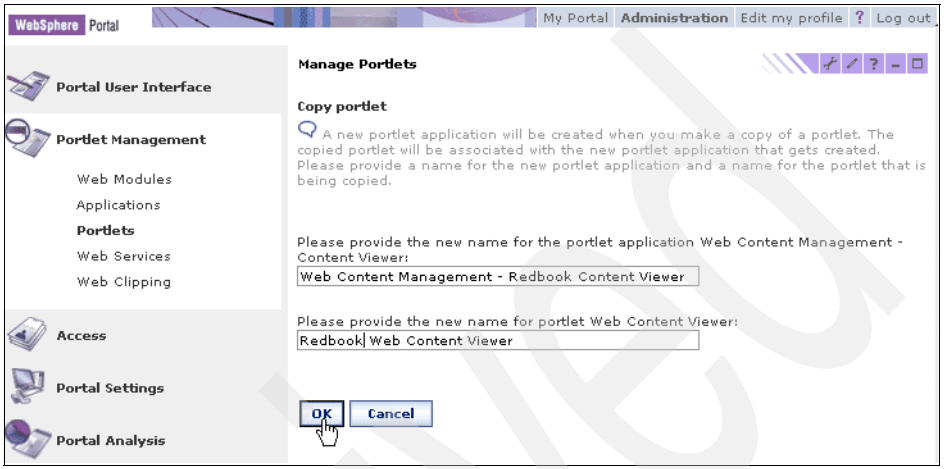


Figure 3-17 Naming the portlet application and portlet

7. Click OK to initiate the copy. When the copy is complete, you receive a confirmation message, as shown in Figure 3-18.

Figure 3-18 Copy portlet confirmation message

Placing the Local Rendering Portlet copy on a page

Now that you have a copy of the Local Rendering Portlet, you can place this copy on a page within our portal. In the following example, we have a page named IBM Redbook Local Rendering, which already exists as a child page to the Web Content Management label. If you wish to follow these instructions exactly as written, create this page prior to continuing with this section. If you already have another page that acts as the home for the portlet, simply use that page whenever we reference the IBM Redbook Local Rendering page.

Note: If you are new to WebSphere Portal and do not know how to create a page, see *IBM WebSphere Portal for Multiplatforms V5.1 Handbook*, SG24-6689, available at:

`http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246689.html?Open`

To deploy the portlet to the IBM Redbook Local Rendering page:

1. In the left hand navigation click **Portal User Interface** → **Manage Pages** to access the Manage Pages GUI.
2. Navigate to the Web Content Management label by clicking **My Portal** and **Web Content Management**. After selecting Web Content Management , you should see a list of the pages under Web Content Management. Click the **Edit Page Layout** that is associated with the IBM Redbook Local Rendering page, as shown in Figure 3-19.

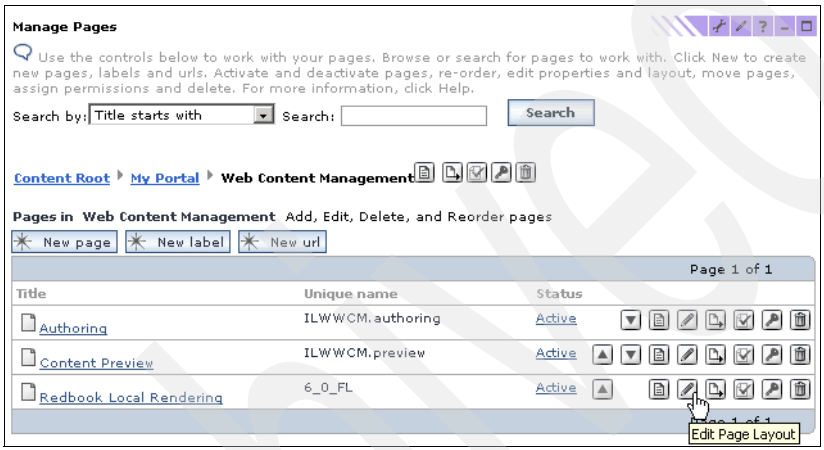


Figure 3-19 Manage Pages GUI - edit page layout

Archived

3. In the Edit Layout screen click **Add Portlets** in the appropriate column container, as shown in Figure 3-20.

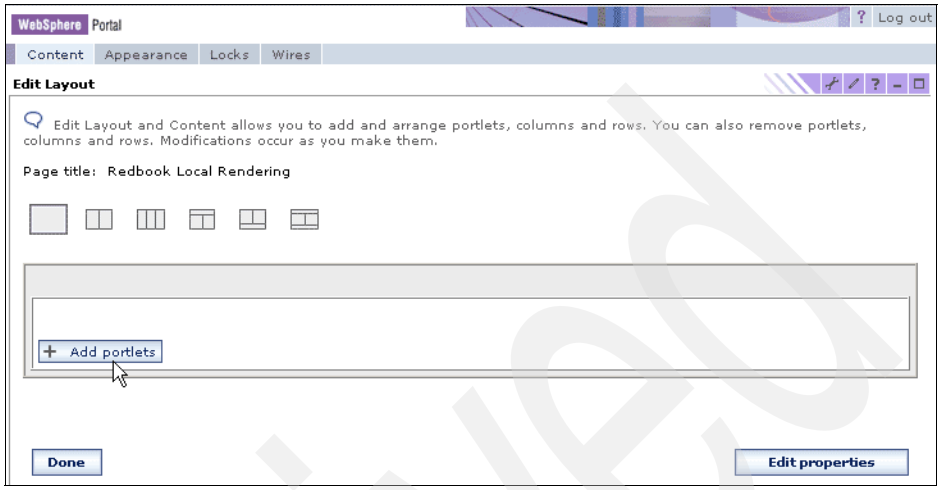


Figure 3-20 Adding a portlet to a column container

4. The next screen allows you to search for the portlet that you wish to deploy to the column container. Search for the portlet by using the parameters *title contains* and *redbook*. You should now see the portlet copy that you created earlier, as shown in Figure 3-21.

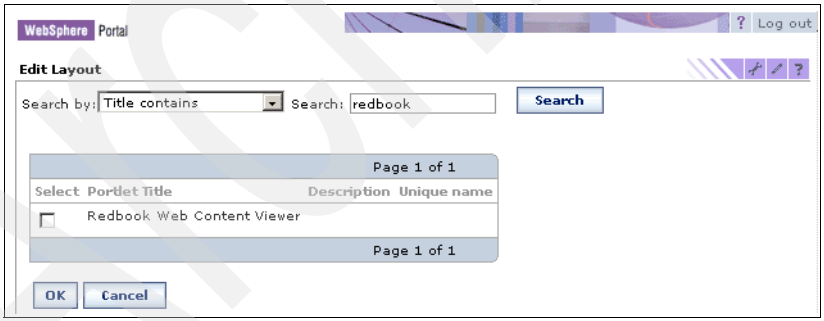


Figure 3-21 Add portlet search results

Archived

5. Select the IBM Redbook Web Content Viewer and then click **OK**. This action returns you to the previous screen. The column container now shows the IBM Redbook Web Content Viewer, as shown in Figure 3-22.

Figure 3-22 Redbook Viewer in column container

6. Click **Done** to return to the Manage Pages GUI.

The final step of deployment is to navigate to the IBM Redbook Local Rendering page and to confirm that the portlet is on the page. If you have successfully deployed the portlet, you should see something similar to that shown in Figure 3-23.

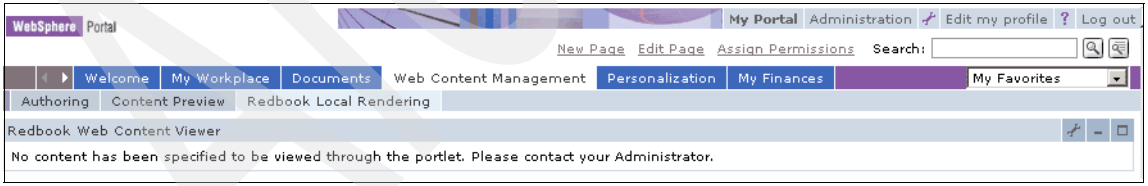


Figure 3-23 Confirming the new portlet is on the IBM Redbook Local Rendering page

As you can see from Figure 3-23, we deployed the portlet successfully to the page, but we have one more process to complete. We must now configure the portlet.

Configuring the Local Rendering Portlet

The final step to successfully deploying the Local Rendering Portlet is to configure the portlet to show IBM Workplace Web Content Management content. In order to perform this task, you must have Administrator rights to the Local Rendering Portlet and to the page on which it is deployed.

Assuming that you have the correct access, click the Configure icon (typically a wrench) in the upper-right corner of the portlet. When you enter configure mode, you see a screen similar to that shown in Figure 3-24.

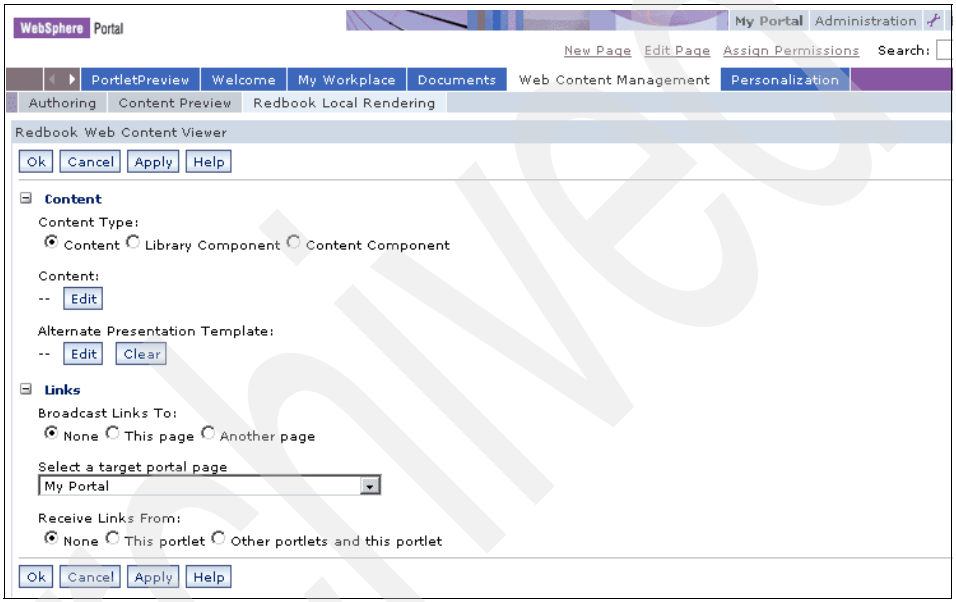


Figure 3-24 Configure mode of the Local Rendering Portlet

The configuration of the Local Rendering Portlet is broken into two sections: the Content section and the Links section.

Content section

The Content section of the Local Rendering Portlet is used to determine the default content that appears in the portlet. You can select from multiple types of IBM Workplace Web Content Management object, including Content and Library Components. When you select the Content Type, the remaining choices in the

Archived

Content Section change based on your selection, as shown in Figure 3-25, Figure 3-26, and Figure 3-27.

Figure 3-25 Configuration options for content type of Content

Figure 3-26 Configuration options for content type of Library Component

Figure 3-27 Configuration options for content type of Content Component

For details on each Content Type, see Chapter 5, “Key concepts and terminology” on page 141 and Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253. For this example, we

Archived

configure the portlet to show the primary page in the News section of the River Bend Coffee and Tea Company Web site (Figure 3-28).

> **Note:** The fictional River Bend site is used as an example site throughout this book as we demonstrate various capabilities of IBM Workplace Web Content Management. It is defined in detail in 7.1, “An overview of the scenario” on page 255.

Figure 3-28 Selecting content from the River Bend Web site

The result of this selection is the River Bend in the News page is rendered in the IBM Redbook Web Content Viewer portlet, as shown in Figure 3-29.

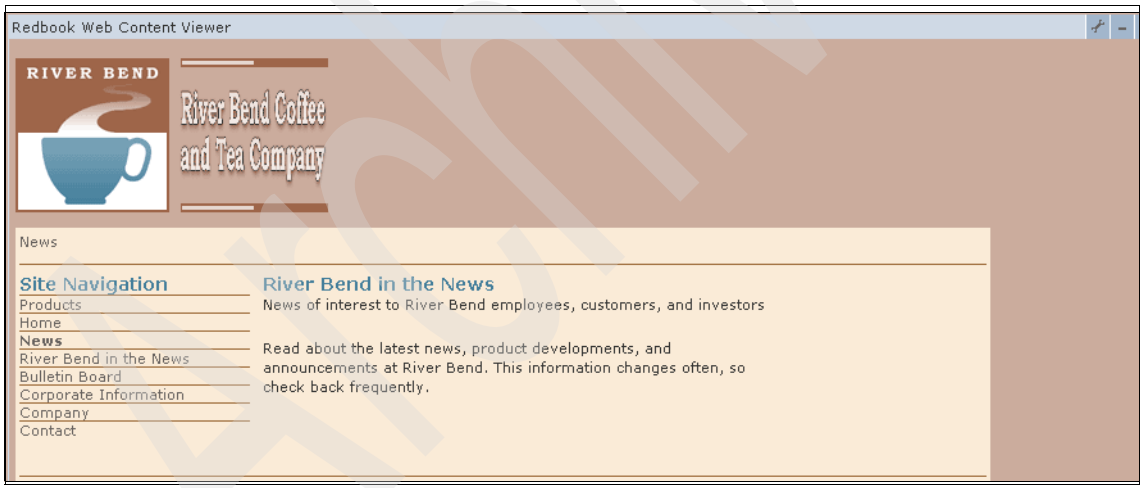


Figure 3-29 River Bend In the News page as shown in the Content Viewer

Archived

Important: As shown in Figure 3-29, you must be careful when displaying IBM Workplace Web Content Management content within WebSphere Portal. You should only use HTML, Java Script, and stylesheet values that are complimentary to your portal themes and skins.

The Alternate Presentation Template option (as shown in Figure 3-28 on page 105) allows you to designate a separate presentation for content when displayed in the portlet. This option allows one piece of content to have multiple presentations based on where the content is displayed. This is one of the most important features of IBM Workplace Web Content Management.

Links section

Multiple rendering portlets can be added to a single WebSphere Portal Server page or a series of pages. Sometimes, it is necessary for different rendering portlets to interact with each other. In these scenarios, you need to modify the Links section of the rendering portlet, as follows:

- Broadcasting Links: The state or context of a rendering portlet is not sent directly from one Portlet to another. Rather, rendering portlets can be configured to broadcast their current state or context to other rendering portlets on the same WebSphere Portal page, or to rendering portlets on other pages. Any information broadcast by a rendering portlet will only be received by rendering portlets configured to receive this information.
- Receiving Links: A rendering portlet can receive information about the state or context of the current Web Content Management Content item or Component being displayed within it, or from Web Content Management Content items or Components displayed within other rendering portlets that are *broadcasting*.

For full details on these settings, refer to the IBM Workplace Web Content Management Information Center.

Restriction: Based on the current behavior of the rendering portlets (both Local and Remote), it is not recommend that you deploy more than three of them on a single WebSphere Portal page.

Archived

3.5.2 Remote Rendering Portlet

If you configure IBMM Workplace Web Content Management on a WebSphere Portal server, you can find the Remote Rendering Portlet at:

```
<WP_Root>/wcm/installableApps
```

The Remote Rendering Portlet is in a Web archive file named ilwwcm-remoterendering-portlet.war.

Deploying the Remote Rendering Portlet

Unlike the Local Rendering Portlet, you typically use the Remote Rendering Portlet on a WebSphere Portal server that is not running IBM Workplace Web Content Management. The process to deploy this portlet involves two major steps:

1. Installing the Remote Rendering Portlet.
2. Deploying the Remote Rendering Portlet to a WebSphere Portal Page.

Installing the Remote Rendering Portlet

> **Important**: Before performing these steps, make sure that the ilwwcm-remoterendering-portlet.war file is in a location to which you can browse from your Web browser.

To install the Remote Rendering Portlet:

1. Log in to WebSphere Portal as an Administrator (for example, member of wpsadmins group).
2. Click the Administration link to access the Administration Console within WebSphere Portal. This is found typically in the upper-right corner of the page if you are using one of the standard WebSphere Portal themes.
3. In the left hand navigation click **Portlet Management** → **Web Modules** to access the Manage Web Modules GUI. From this screen, click **Install** to start the installation process.

Archived

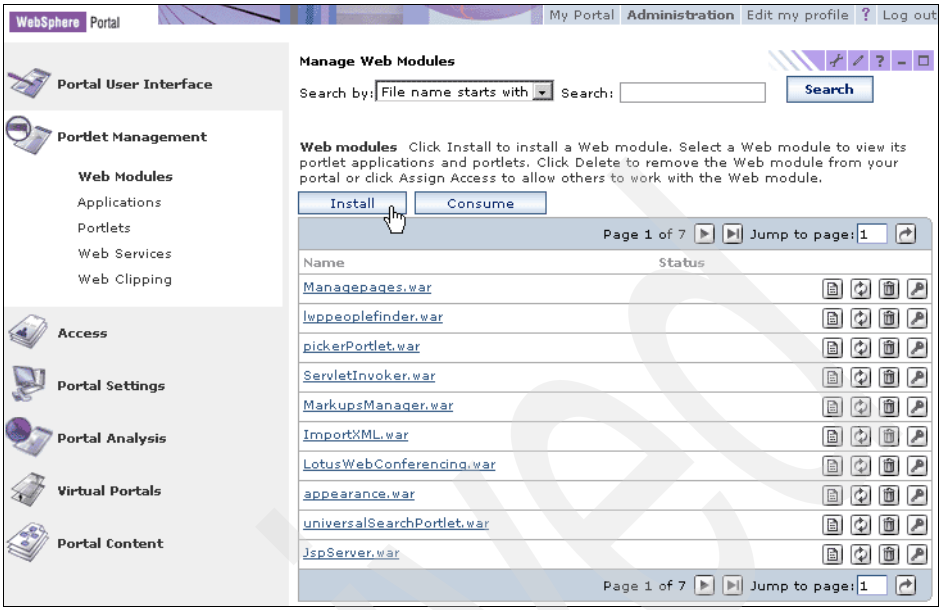


Figure 3-30 Portlet Management Web Modules GUI

4. The next screen prompts you for the location of a portlet WAR file. Browse to the location of the ilwwcm-remoterendering-portlet.war file. Select this file, and click **Next**.

Figure 3-31 Installing a Web Module - Selecting the correct WAR file

5. You see a confirmation screen that shows you the names of the Portlet Applications and Portlets that you are about to install. (Note that the name of the portlet is *Remote Web Content Viewer*. You need this name when you deploy the portlet to a page).

Figure 3-32 Installing a Web Module - WAR contents

When the installation completes, you return to the Manage Web Modules GUI with a small confirmation that the portlet was installed.

Figure 3-33 Installing a Web Module - Confirmation

Now that you have installed the portlet you can deploy it to a portal page.

Deploying the Remote Rendering Portlet

The process of deploying the Remote Rendering Portlet is identical to that of the Local Rendering Portlet. The only difference is that the portlet has a different name. So, you need to search for the portlet using this name. Refer to “Deploying the Local Rendering Portlet” on page 95 on deploying the local portlet for instructions.

Configuring the Remote Rendering Portlet

Configuring the Remote Rendering Portlet is very similar to configuring the Local Rendering Portlet. The only real difference is that the Remote Rendering Portlet requires some additional information. These differences are due to the fact that this portlet — unlike the Local Rendering Portlet — interacts with the IBM Workplace Web Content Management much as a Web browser would.

The Remote Rendering Portlet makes a URL connection to IBM Workplace Web Content Management to get content. In order to facilitate this interaction, there are two sections in the configure mode of the Remote Rendering Portlet that are not in the Local Rendering Portlet: credential section and settings section.

Archived

Important: The Remote Rendering Portlet has Content and Links sections that are identical in function to those in the Local Rendering Portlet. For a review of these sections, refer to “Configuring the Local Rendering Portlet” on page 103.

Credential section

By default, the Remote Rendering Portlet uses the Portal user's logon to access content on the remote IBM Workplace Web Content Management server. Alternately, it can use a valid Credential Vault Slot name instead of the user's logon.

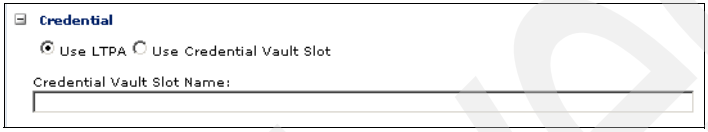


Figure 3-34 Credential section

In most cases, you only need to override a Portal user's logon if the Portal Server used different logons from IBM Workplace Web Content Management or if you wanted all users to have the same credentials for a particular Remote Rendering Portlet. You can use the following to override a Portal user’s logon:

- Use LTPA: IBM Workplace Web Content Management use the Portal user's logon to determine access rights to a Remote Rendering Portlet. It is important to note that for LTPA to work, your WebSphere Portal and remote IBM Workplace Web Content Management servers must share the same LTPA token. This seems like an obvious requirement, but we have seen many cases where the IBM Workplace Web Content Management servers are not configured to share an LTPA token with the portal servers.

  See the WebSphere Portal Information Center for further information about LTPA.

- Use Credential Vault Slot: You can use a valid Credential Vault Slot name instead of the user's logon. You need to enter a valid Credential Vault Slot in the field. The vault slot must already exist before you configure this portlet.

Archived

Settings section

In order to make a remote connection to the IBM Workplace Web Content Management server, the Remote Rendering Portlet needs some basic information regarding the IBM Workplace Web Content Management server and its configuration. The settings section is where you set up these values, as shown in Figure 3-35 and Figure 3-36.

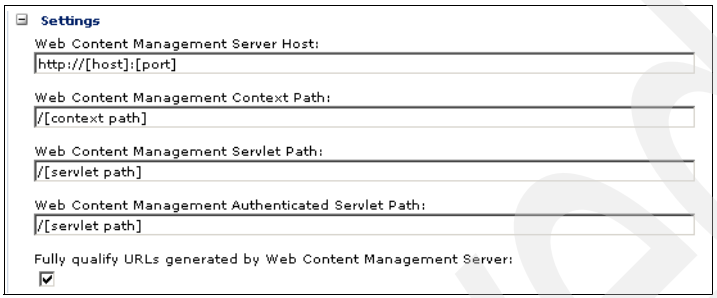


Figure 3-35 Default values for Settings section

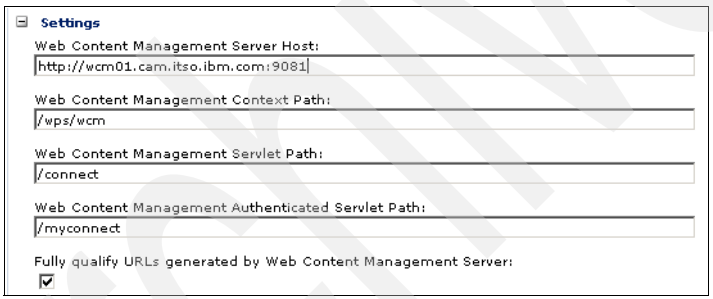


Figure 3-36 Values used in ITSO Lab for Settings section

The IBM Workplace Web Content Management Information Center provides a very good description of these values and what the typical defaults are for each value.

Tip: You should fill out the Settings and Credential sections first when configuring the Remote Rendering Portlet. After entering the appropriate values, click **Apply**. When this is completed, you should be able to fill in the other sections.

Restriction: Based on the current behavior of the rendering portlets (both Local and Remote), it is not recommend that you deploy more than three of these portlets on a single WebSphere Portal page.

Archived

3.6 Post-installation configuration

When you complete the basic installation of IBM Workplace Web Content Management, there are some optional configuration settings you might wish to modify. This section looks at some of these options.

3.6.1 E-mail

IBM Workplace Web Content Management provides for basic SMTP mail routing. This capability is especially important if you wish to use e-mail messages as the primary method for communicating with content creators and approvers as content goes through its workflow approval cycle.

Basic mail routing configuration

This section describes how to configure this e-mail function. Within the <*WP_Root*>/wcm/config/connect.cfg set the following `<connector>` value:

```
<MailConnector>
      <DefaultSMTPServer value=mail.yourmailserver.com />
      <DefaultFromAddress value='admin@yourmailserver.com' />
      <DefaultReplyToAddress value='admin@yourmailserver.com' />
      <ConnectionManager>
         <MaxConnectionPools value=1000 />
         <MaxConnectionsPerPool value=64 />
      </ConnectionManager>
</MailConnector>
```

In this example, *`mail.yourmailserver.com`* is the name of your SMTP mail server. Change the `FromAddress` and `ReplyToAddress` values as appropriate.

> **Note:** The SMTP server value does not have open and close quotation marks around it.

Configuration for fatal mail alerts

In addition to the basic mail routing configuration, you can configure fatal mail alert settings so that an e-mail is sent to an administrator when appropriate. The settings are as follows:

```
<Mail>
   <FatalMail>
      <FatalMailMgr value='Framework@yourServer.com' />
      <FatalMailFrom value='Framework@yourServer.com' />
   </FatalMail>
</Mail>
```

Within the <*WP_Root*>/wcm/config/aptrixjpe.properties file, change the following setting:

```
default.system-email=admin@yourmailserver.com
```

In this example, *admin@yourmailserver.com* is the value used by the e-mail action in workflow. This is the default e-mail address to be used as the from address when a workflow action is executed.

3.6.2 Logging levels

IBM Workplace Web Content Management logs various activities to the following log files:

- <*WP_Root*>/wcm/connect/log/connect.log
- <*WP_Root*>/wcm/connect/log/debug.log
- <*WP_Root*>/wcm/connect/log/error.log

If you experience issues with IBM Workplace Web Content Management, a good first step in debugging is to look at these log files. One nice feature of IBM Workplace Web Content Management logging is that you can change the level (verbosity) of the logging by modifying settings in the connect.cfg file, as shown in Example 3-4.

Example 3-4 Default values for the logging options in connect.cfg

```
<!-- Log manager configuration -->
<LogManager>
   <ErrorLog>
      <File LogFile="../connect/log/error.log" FlushLog=false
      Buffered=false TraceTime=true TraceDate=true
      TraceLevel=3 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=true
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>
```

As you can see from the sample file, you have the ability to set a logging level for each of the log files. The trace levels can be from Level 0 to Level 5 and have the following meanings:

- Level 0: No output except broad details regarding critical system exceptions (errors).
- Level 1: All prior level logging plus general system events (start up, shut down, and so forth), and all other system exceptions.
- Level 2: All prior level logging plus common events such as requests from clients.
- Level 3, Level 4, and Level 5: All prior level logging plus various levels of programmer-readable debug. These types of log contents are not necessarily readily understandable by anyone outside of the IBM Workplace Web Content Management development team. These levels should not be used unless a Web Content Management solution is not functioning properly and the reasons behind this need to be determined (debugging).

Restriction: To use logging levels above Level 1, you need to install the IBM Workplace Web Content Management debug .jar files. These files are available on request from IBM Workplace Web Content Management support. They do not ship with IBM Workplace Web Content Management.

3.6.3 Caching

Caching is typically used to improve site performance. IBM Workplace Web Content Management provides multiple techniques for caching site content. While caching is a configuration process, it is *not* recommended that you enable caching until you build out your Web site and get a baseline for performance without caching.

Chapter 8, “Caching and pre-rendering considerations” on page 347 presents a detailed discussion on caching.

3.7 Static pre-rendering

In a typical configuration, the IBM Workplace Web Content Management engine is part of the production runtime environment. When people request content from IBM Workplace Web Content Management, the IBM Workplace Web Content Management Content Server serves the request in real time. However, there are some situations where a site does not require dynamic behaviors, such as Personalization or real-time access to databases. In situations such as these,

IBM Workplace Web Content Management provides the ability to pre-render an entire site to a set of static HTML files.

While you lose the IBM Workplace Web Content Management dynamic capabilities if you chose pre-rendering, you realize the following benefits:

- Cost Savings: If your site is pre-rendered you no longer need runtime IBM Workplace Web Content Management licenses. Instead, your pre-rendered site can be placed on an HTTP Server (for example, IBM HTTP Server or Apache).
- Performance Improvement: You see considerable performance improvements because rendering static HTML files on an HTTP server always perform better than assembling content dynamically at request time.

This section discusses how to configure IBM Workplace Web Content Management to pre-render a site and what the rendering engine produces as its output.

3.7.1 Enabling the correct caching subsystem

The IBM Workplace Web Content Management application contains two subsystems for static caching of content. To pre-render a site you must enable the correct subsystem.

Important: The two caching subsystems are mutually exclusive. You should not enable both of them at the same time!

A default installation of IBM Workplace Web Content Management does not enable either of the caching modules. Before you can use the static pre-rendering in IBM Workplace Web Content Management, you must turn on the cacher and enable the correct caching module.

Enabling the cacher

To enable the IBM Workplace Web Content Management Cacher, you make one change to the connect.cfg file that is located in the <*WP_Root*>/wcm/config directory.

Archived

In the ModuleConfig section of the connect.cfg file, there is a single XML tag for the cacher. By default, this tag is commented out. Remove the comment tags from the `<Cacher>` tag, as shown in Example 3-5.

Example 3-5 Fragment of the connect.cfg file, enabling the <Cacher> tag

```
...
<ModuleConfig>
   <AJPE>
      <Properties value="../config/aptrixjpe.properties" />
      <AptrixHome value="../" />
      <DependentModules>
         <EnsureUserManagement />
         <Syndication />
         <AJPERES />
         <Default />

         <!-- Uncomment when cacher is enabled and not default -->
         <Cacher />
```

Enabling the correct caching module

When we know the cacher is enabled we must choose the appropriate caching module for pre-rendering, as shown in Example 3-6. The two choices for caching module appear next to each other in the connect.cfg file that is located in the <*WP_Root*>/wcm/config directory.

Example 3-6 Fragment of the connect.cfg file, enabling the correct cacher module

```
...
<!-- AJPE -->

      <!-- Renderer Module -->
      <Default class=com.aptrix.pluto.renderer.RendererModule remoteAccess=true
autoLoad=false />

      <!--
           Cacher Module (as default)
           ensure that RendererModule is commented out
      -->
      <!--
      <Default class=com.aptrix.cacher.CacherModule remoteAccess=true
autoLoad=false />
      -->

      <!-- Cacher Module (stand alone) ensure that
ModuleConfig/AJPE/DependentModules/Cacher is uncommented -->
```

```
      <Cacher class=com.aptrix.cacher.CacherModule remoteAccess=true
autoLoad=false />

...
```

The IBM Workplace Web Content Management development team was nice enough to provide comments next to each of the cacher module tags to guide us towards the correct selection; just make sure you only have one of these two choices uncommented!

3.7.2 Configuring the <Cacher> configuration options

The `<Cacher>` subsection of `<ModuleConfig>` is where you can configure the specific settings you wish to use for the pre-rendering process. These settings are generally divided into two categories:

- **When** the Cacher runs
- **How** the Cacher runs

The settings for configuring the Cacher are located in the connect.cfg file, as shown in Figure 3-37.

```
<Cacher>
   <CacherClass value="com.aptrix.cacher.Cacher" />
   <DestDir value="D:/WEBSPH~1/PORTAL~1/wcm/ilwwcm/cacher" />
   <TempDir value="D:/WEBSPH~1/PORTAL~1/wcm/ilwwcm/cacher/temp" />
   <Delay value="1" />
   <BusyDelay value="5" />
   <BusyStart value="9:00 am" />
   <BusyEnd value="5:00 pm" />
   <OverwriteCache value="true" />
   <DefaultContentName value="index.html" />
   <!-- choices are Anonymous, Administrator or a specific user or group common name -->
   <RendererUser value="Anonymous" />
   <!-- <Password value="put_your_password_here" /> -->
   <Task>
      <CacherUrl value="http://wcm02.cam.itso.ibm.com:9081/wps/wcm/connect/" />
      <ServletPath value="/connect" />
      <Sites value="RiverBend" />
      <Interval>
         <Recurrence value="300" />
         <StartDelay value="1" />
      </Interval>
      <Scheduled>
         <Times value="3:00 am" />
      </Scheduled>
   </Task>
</Cacher>
```

Figure 3-37 Sample configuration of the <Cacher> section in connect.cfg file

The `<Task>` section determines when the Cacher will run and what Sites will be pre-rendered during a rendering cycle. If you elect to have pre-rendering run on a regular schedule, the `<Interval>` and `<Scheduled>` tags are used to tell the rendering engine when to run.

Similar to some of the configuration settings in the Remote Rendering Portlet, the remaining tags in the `<Task>` section tell the Cacher how it can "connect" to the

site. A standard IBM Workplace Web Content Management installation typically gives correct values to the `<CacherUrl>` and `<ServletPath>` tags. You always want to modify the `<Sites>` tag to contain applicable values for your installation. The `<Sites>` tag expects a comma separated list of sites that should be pre-rendered each time the rendering process runs.

While the remaining tags in the `<Cacher>` section are (for the most part) self explanatory. The following tags require your attention:

- `RendererUser`: This tag tells the pre-rendering engine what user should be used to render the content. Because the entire site is rendered to *flat* HTML files, you do not have dynamic behavior in the resultant pages. However, you can choose to render with a specific user ID, and that decision can effect what is rendered based on the security within the site content.
- `Password`: If you choose to use an ID other than anonymous, you must tell the rendering engine what the password is for the ID that you provide. It is important to note that, by default, the connect.cfg file does not have this tag in the `<Cacher>` section. This tag is added manually after installation.
- `OverwriteCache`: This tag effects how rendered content is created and when the last set of rendered content is overwritten. When set to `true`, the existing (last rendered) content is deleted from the `<DestDir>` before the new rendering process starts. In this scenario, the new content is rendered directly to the `<DestDir>`.

  When set to false, the existing (last rendered) content is deleted by the renderer only `after` the new rendering process is completed. The newly rendered content is stored in `<TempDir>` and only moved to `<DestDir>` when rendering is completed. In addition, if you set this to `false`, the render engine saves the prior version of content in a folder called `<site_name>`-old. This setting provides a very basic ability to roll back one version if the rendered content proves to have problems.
- `DefaultContentName`: When content is created, the rendering engine uses this value to determine the name of the HTML page that it creates as the default content in each Site Area. For most HTTP servers, such as IBM HTTP Server or Apache, you do not need to change the default of index.html.

If you plan on pre-rendering content, you should become very familiar with this section of the connect.cfg file and its options. For details on tags that this section did not cover, see the IBM Workplace Web Content Management Information Center.

After you make the appropriate configuration changes, restart the WebSphere Portal and IBM Workplace Web Content Management server. Upon restart, the pre-rendering engine should start to pre-render the site.

3.7.3 Manually running the pre-rendering engine

Assuming pre-rendering is properly configured, it is possible to initiate the pre-rendering engine by issuing a single URL in your Web browser:

```
http://[HOST]:[PORT]/wps/wcm/connect/?MOD=Cacher&SRV=cacheSite&Site=
[SITENAME]
```

For example:

```
http://wcm01.cam.itso.ibm.com:9081/wps/wcm/connect/?MOD=Cacher&SRV=
cacheSite&Site=RiverBend
```

When you enter this URL, you are prompted for a user name and password. The person running this task *must* be in the wmcadmins group. Note that when using this URL you can only render a single site at a time.

Note: In IBM Workplace Web Content Management 5.1 (and 2.5), there is no message sent to the browser window that tells you when rendering is complete. If you have `<OverwriteCache>` set to `false`, you can watch the `<TempDir>` to determine when the rendering is completed. While rendering is in process, the site content is in `<TempDir>`. After rendering is complete, the `<TempDir>` is empty.

3.7.4 Known issues with pre-rendering in IBM Workplace Web Content Management 5.1

This section discusses the two known issues with the static pre-rendering functionality in IBM Workplace Web Content Management 5.1 at the time of this writing.

Manual versus automated rendering

As discussed earlier, you have the choice of pre-rendering a site automatically or manually via a URL. You can even combine the two as needed to ensure that the site content is as current as needed. However, the URLs that are generated to access IBM Workplace Web Content Management resources (for example, File and Image Components) are different depending on how you invoke the rendering process.

Relative paths ... A moving target

In all cases, the rendering engine builds relative paths to images and other resources that are needed by each HTML page. When you schedule the rendering process, these relative paths start with /wps/wcm/connect. However, when you run the rendering servlet manually, you must first authenticate to the

WebSphere Portal environment, which causes all relative URLs to start with /wps/wcm/myconnect. These differing paths could lead to issues when you deploy the pre-rendered content to your HTTP server.

Solution to this issue

IBM HTTP Server (and Apache) can solve this problem by enabling the rewrite module and by setting up two simple rewrite rules. The rewrite module is included with the most recent version of IBM HTTP Server. So, all you need to do is enable the module and create two rules as follows:

- Enable the rewrite module: The latest release of IBM HTTP Server includes the rewrite module by default. The configuration line needed to load the module is in the <*IHS_Root*>/conf/httpd.conf file, but it is commented out. To have the rewrite module load, uncomment the following line:

  ```
  LoadModule rewrite_module modules/mod_rewrite.so
  ```

- Create rewrite rules: When the rewrite engine is running, give it some instructions as to how it should function. The process is very simple assuming that you know the basics of regular expressions. The most important of the directives is `RewriteEngine ON`. Without this directive, the engine is not running even though the module is loaded.

  After the rewrite engine is configured to run, you need to create rules that instruct the engine when to intervene. You can place any number of RewriteRule directives in the httpd.conf file to provide scenarios in which rewriting should take place.

  Each one of these rules uses Regular Expressions to determine what URL should be rewritten and how it should be modified. The rules shown in Figure 3-38 remove the strings `wps/wcm/connect` or `wps/wcm/myconnect` from any URL requested from the IBM HTTP Server.

```
# Rewrite Engine Test
RewriteEngine ON
RewriteLog "C:/Program Files/Apache Group/Apache2/htdocs/rewrite.log"
RewriteRule ^(.*)/wps/wcm/myconnect/(.*)$  $1/$2 [R,L]
RewriteRule ^(.*)/wps/wcm/connect/(.*)$  $1/$2 [R,L]
#RewriteLogLevel 3
```

Figure 3-38 Rewrite module rules to fix path issue

IBM Workplace Web Content Management and pre-rendered URLs

Now that you know how to create rewrite rules for these changing URLs, let us look at how you can use these rules when deploying a pre-rendered site to IBM HTTP Server.

Note: Resources (files and images) are a type of component within IBM Workplace Web Content Management. For a full explanation of these resources, see Chapter 5, “Key concepts and terminology” on page 141.

The default URLs that IBM Workplace Web Content Management generates for links, images, files, and other IBM Workplace Web Content Management managed resources are relative URLs. They always start with one of the two strings that you are trying to rewrite. For example, the following might be a reference to a stylesheet. :

```
href="/wps/wcm/myconnect/MySite/resources/file/eb5b194f9814416/std.css"
```

Without any rewrite rules, you have to place the IBM Workplace Web Content Management rendered files one of the following directories:

```
<IHS_Root>\htdocs\wps\wcm\connect\<Site>
<IHS_Root>\htdocs\wps\wcm\myconnect\<Site>
```

Unfortunately, the correct directory is dependent on how you render the site (manually or scheduled), which means the URLs could change any time you render if you use the opposite approach. With our rewrite rules, you can make IBM HTTP Server serve the content properly regardless of how it is rendered.

Using the rewrite rules shown earlier, any IBM Workplace Web Content Management relative URLs would be shortened and start with the site name. Thus, our example css reference is:

```
"/MySite/resources/file/eb5b194f9814416/std.css"
```

On the IBM HTTP Server, you can now install any copy of the rendered site directly into the document root directory (for example, <*IHS_Root*>\htdocs\<*Site*>) regardless of how the files were rendered.

Resources do not render

During a proper pre-rendering process, there are two phases that the pre-rendering engine performs:

- Create HTML Files: This phase creates the HTML files for all site content that is visible to the account specified in the `<RendererUser>` field within connect.cfg.
- Create a resources folder: Because the pre-rendering engine creates the HTML files, it keeps track of the resources (files and images to be specific) that are referenced by the HTML pages. After all the site content is pre-rendered, this phase creates a resources folder and places any required resources with the folder. This step is required as any given site might only need a fraction of the resources in IBM Workplace Web Content Management

so it would be inappropriate to include all resources for any one pre-rendered site.

Unfortunately, a feature in IBM Workplace Web Content Management 5.1.0 causes IBM Workplace Web Content Management to not create the resources folder.

> **Note:** The pre-rendering process works properly in IBM Workplace Web Content Management 2.5. The resources folder is properly created.

The only workaround at this time is to place the required resources on the HTTP server manually and to use an additional rewrite module rule to create an alias to the resources. You can find all non-secured resources at:

```
<WAS_Root>/installedApps/node_name/ilwwcm.ear/ilwwcm.war/resources
```

You can copy this entire folder to the Web server and then set up a rewrite rule to map URLs that contain resources to this folder. The primary issue with this approach is that the <*WAS_Root*>/installedApps/node_name/ilwwcm.ear/ilwwcm.war/resources folder would contain all non-secured resources regardless of the sites that need them. If this is a concern, you need to determine which files were needed for the pre-rendered site manually.

3.8 Removing IBM Workplace Web Content Management

In the event that you need to remove the IBM Workplace Web Content Management application from a WebSphere Portal server, the **remove-wcm** configuration task removes the entire IBM Workplace Web Content Management application from the WebSphere Portal installation.

To run this task:

1. Stop WebSphere Portal.
2. Open a command window or command prompt.
3. Navigate to the <*WP_Root*>/config directory.
4. Run the following script:

   ```
   wpsconfig remove-wcm
   ```

5. Start WebSphere Portal.

When the task is complete, the IBM Workplace Web Content Management application is removed from your system.

Note: The IBM Workplace Web Content Management configuration files are also removed when running this configuration task. If you plan g to reinstall IBM Workplace Web Content Management, back up these files first.

Archived

Archived

4

Overview of IBM Workplace Web Content Management 2.5

This chapter discusses IBM Workplace Web Content Management 2.5. It highlights its positioning, its target audience, the subtle differences in the installation process, and its functionality and features. It includes the following sections:

- Introduction to IBM Workplace Web Content Management 2.5
- Functional differences between IBM Workplace Web Content Management 2.5 and 5.1
- Installing IBM Workplace Web Content Management 2.5
- Configuring IBM Workplace Web Content Management 2.5

© Copyright IBM Corp. 2006. All rights reserved.

4.1 Introduction to IBM Workplace Web Content Management 2.5

IBM Workplace Web Content Management 2.5 provides a major architectural change from earlier editions. Unlike earlier versions, IBM Workplace Web Content Management 2.5 runs entirely within WebSphere Portal or Lotus Workplace. IBM Workplace Web Content Management 2.5 comes in two versions:

- IBM Workplace Web Content Management 2.5
- IBM Workplace Web Content Management 2.5, Standard Edition

The code base for these versions is identical. However, the Standard Edition contains several significant licensing limitations, as described in 1.6, “Licensing” on page 22. Additionally, the Standard Edition is also the version that is supplied with IBM Workplace Collaboration Services 2.5.

4.2 Functional differences between IBM Workplace Web Content Management 2.5 and 5.1

While the two versions of IBM Workplace Web Content Management 2.5 are functionally identical, the same cannot be said for IBM Workplace Web Content Management 2.5 and IBM Workplace Web Content Management 5.1. IBM Workplace Web Content Management 2.5 is supported for WebSphere Portal 5.0.2.2 and later, while IBM Workplace Web Content Management 5.1 is support for WebSphere Portal 5.1.

This section describes the most notable differences between these versions. To find the most up-to-date information about what is supported in these versions of IBM Workplace Web Content Management, consult the Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

4.2.1 Supported browsers

The IBM Workplace Web Content Management Information Center provides a complete list of supported browsers. The supported browsers include various versions of Internet Explorer, Mozilla, Firefox, Netscape, and so forth. However, not all functions are supported in all browsers. We strongly recommended that you review the Information Center for your specific version of IBM Workplace Web Content Management to confirm which browsers are supported.

4.2.2 Supported Java Virtual Machines

The IBM Workplace Web Content Management Information Center also provides a complete list of supported Java Virtual Machines (JVM™). The Authoring Portlet relies on the JVM as does the Configure mode of both rendering portlets. We strongly recommended that you review the Information Center for your specific version of IBM Workplace Web Content Management to confirm which JVMs are supported.

4.2.3 Portal Document Manager integration

IBM Workplace Web Content Management 5.1 provides a level of integration with the Portal Document Manager in WebSphere Portal 5.1. Portal Document Manager was redesigned significantly in WebSphere Portal 5.1 and the underlying code for Portal Document Manager in WebSphere Portal 5.0 and late is quite different.

IBM Workplace Web Content Management 2.5 does not support integration with the Portal Document Manager.

4.2.4 ODC (Rich Text) editors

The Rich Text editor in IBM Workplace Web Content Management 2.5 does not have the same capabilities as the one that is provided in IBM Workplace Web Content Management 5.1. The OdcEditor.jsp enables only functions that are available in version 5.0.

4.2.5 Fix packs and iFixes

IBM releases fix packs and other smaller fixes (iFixes) as needed for IBM products as follows:

- Fix packs

  Because IBM Workplace Web Content Management 5.1 is integrated with WebSphere Portal 5.1, fix packs for the two products are shipped together in an integrated fashion.

  Because IBM Workplace Web Content Management 2.5, Lotus Workplace, and WebSphere Portal 5.0 and later are separate products, fix packs for each are released separately.

- iFixes

  iFixes are smaller patches that are made available to resolve specific product issues between fix pack releases.

Archived

Important: iFixes for IBM Workplace Web Content Management 5.1 are not compatible with IBM Workplace Web Content Management 2.5 or vice versa.

4.2.6 Portal Personalization (available with 5.1.0.1 and later)

A Personalization component has now been included to enable content based on WebSphere Portal Personalization rules to be rendered as IBM Workplace Web Content Management Components.

Note: For a complete listing of functionality provided with WebSphere Portal 5.1.0.1, see:

http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcm5101/$File/wcm-5-1-0-1-releasenotes.htm

4.3 Installing IBM Workplace Web Content Management 2.5

Unlike IBM Workplace Web Content Management 5.1, IBM Workplace Web Content Management 2.5 is not included in the installation of WebSphere Portal 5.0.2.2 and 5.0.2.3 or IBM Workplace Collaboration Services 2.5. Instead, a separate installation process is required to add the IBM Workplace Web Content Management 2.5 source code to any of the supported environments.

This section provides the general steps that are required to install IBM Workplace Web Content Management 2.5.

Attention: These installation instructions assume that you have a working (and supported) WebSphere Portal or IBM Workplace server.

Archived

4.3.1 Installing IBM Workplace Web Content Management 2.5 on Windows 2000 Advanced Server

This section discusses the installation process for adding IBM Workplace Web Content Management 2.5 to an existing WebSphere Portal 5.0.2.2 server.

> **Attention:** Before you begin installing IBM Workplace Web Content Management, make sure your user repository (LDAP) contains a group called *wcmadmins*. This group is the administrative group for the IBM Workplace Web Content Management application.
>
> If you use the WebSphere Portal built-in Cloudscape database as your user repository, the installation does this step for you automatically. However, we do not recommend this user repository for a production environment.

To install the IBM Workplace Web Content Management 2.5 software, complete the following steps:

1. Open a command prompt to the root directory of the IBM Workplace Web Content Management distribution and end the following command:

   ```
   install.bat
   ```

2. Select the language to use for the installation and click **OK** (Figure 4-1).

Figure 4-1 Installation language selection

   The IBM Workplace Web Content Management installation program Welcome panel displays.

3. Click **Next** to continue.

Archived

The software license agreement for IBM Workplace Web Content Management appears (Figure 4-2).

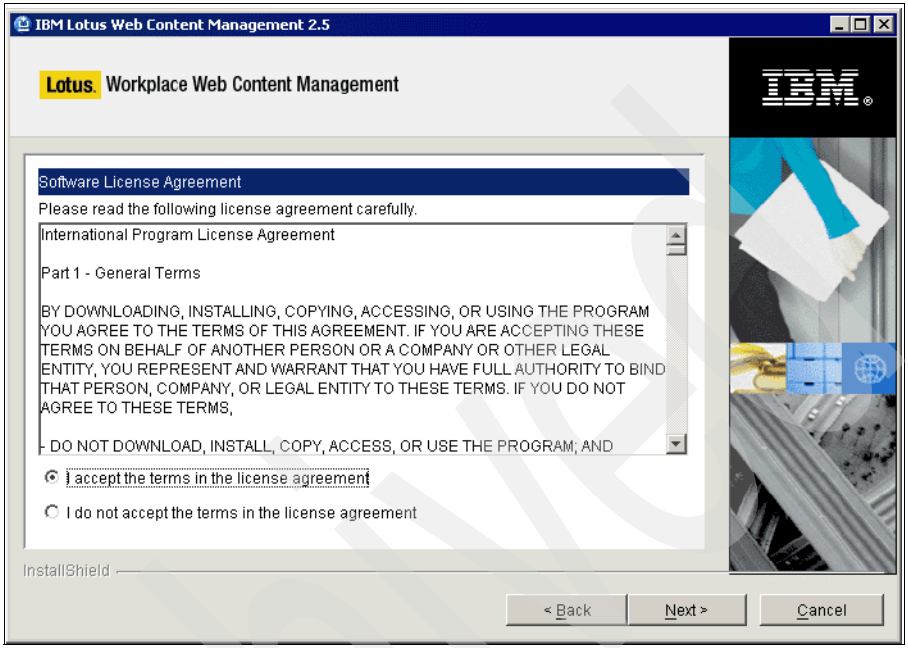


Figure 4-2 Installation license agreement

4. Agree to the license and click **Next**.

Archived

The installation program searches for a base installation of WebSphere Portal or Lotus Workplace. The screen populate this root directory value automatically, as shown in Figure 4-3.

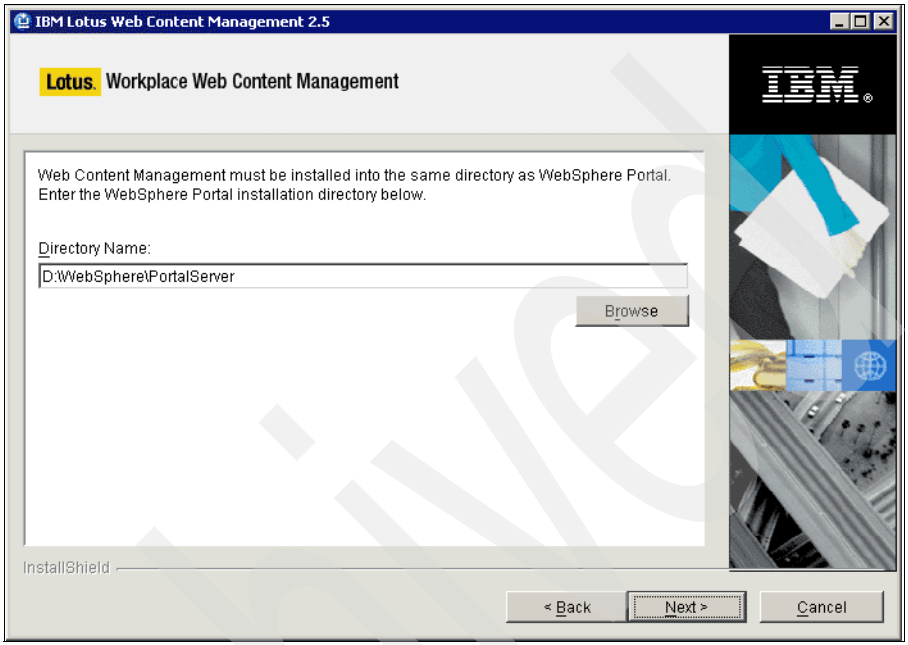


Figure 4-3 Installation install root

5. Click **Next**.

Archived

In order to properly install IBM Workplace Web Content Management, the installation program needs to know if WebSphere Application Server security is enabled (Figure 4-4). If your installation of WebSphere Portal or Lotus Workplace is configured to use an LDAP, the answer to this question is Yes.

6. Select **Yes** and click **Next**.

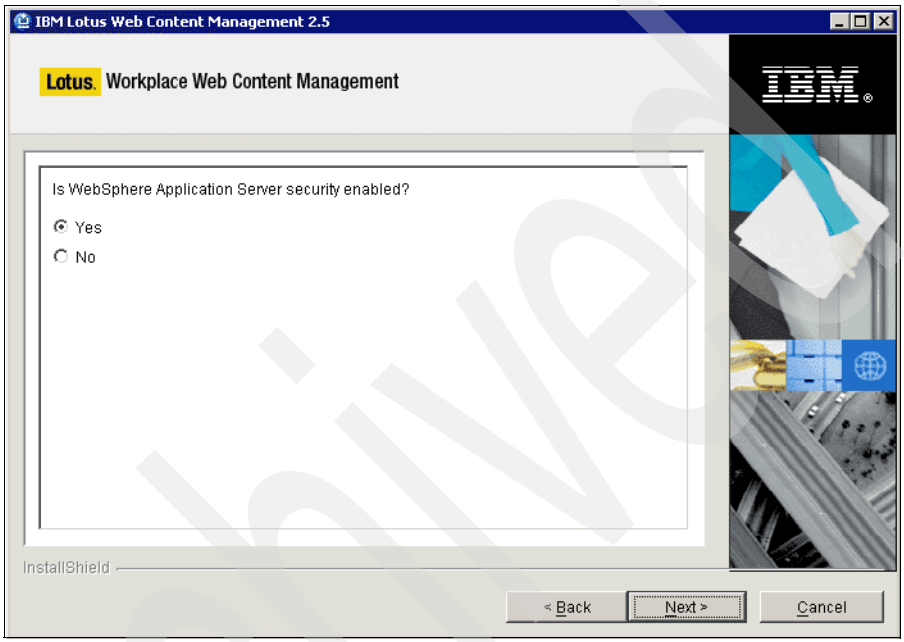


Figure 4-4 WebSphere Application Server security prompt

Archived

Assuming that you have WebSphere Application Server security enabled, are prompted for the administrative user of the WebSphere Application Server (not the WebSphere Portal server), as shown in Figure 4-5. When entering the administrative user make sure you use a fully qualified name.

7. Enter the appropriate values for your installation and click **Next**.

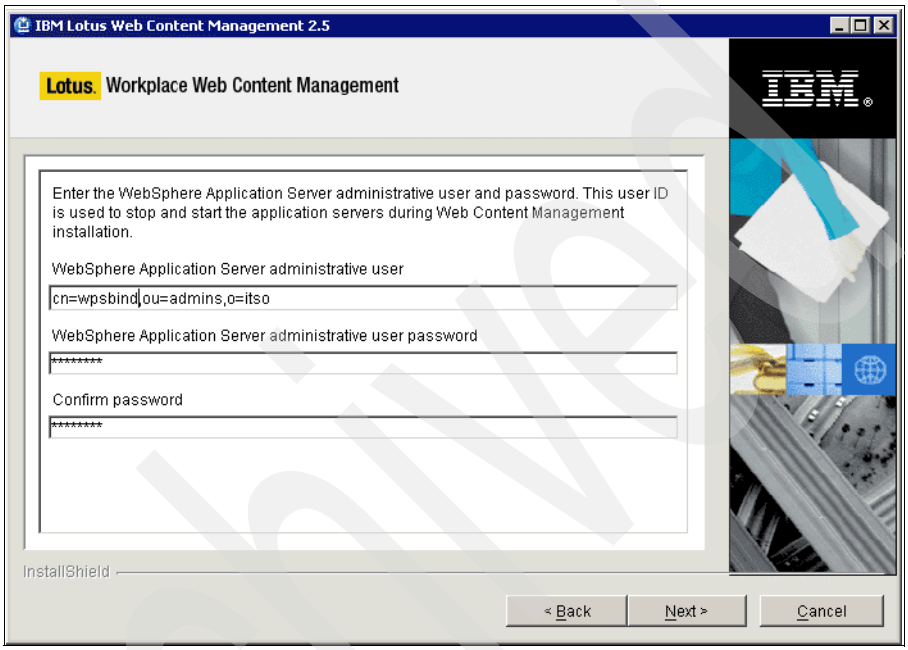


Figure 4-5 Installation WebSphere Application Server administrator details

Archived

Assuming that you have WebSphere Application Server security enabled, you are prompted for an administrative user of WebSphere Portal (not the WebSphere Application server), as shown in Figure 4-6. Unlike the administrative user of the application server, when entering the portal administrative user, you can use the common or short name (for example, `wpsadmin`).

8. Enter the appropriate values for your installation and click **Next**.

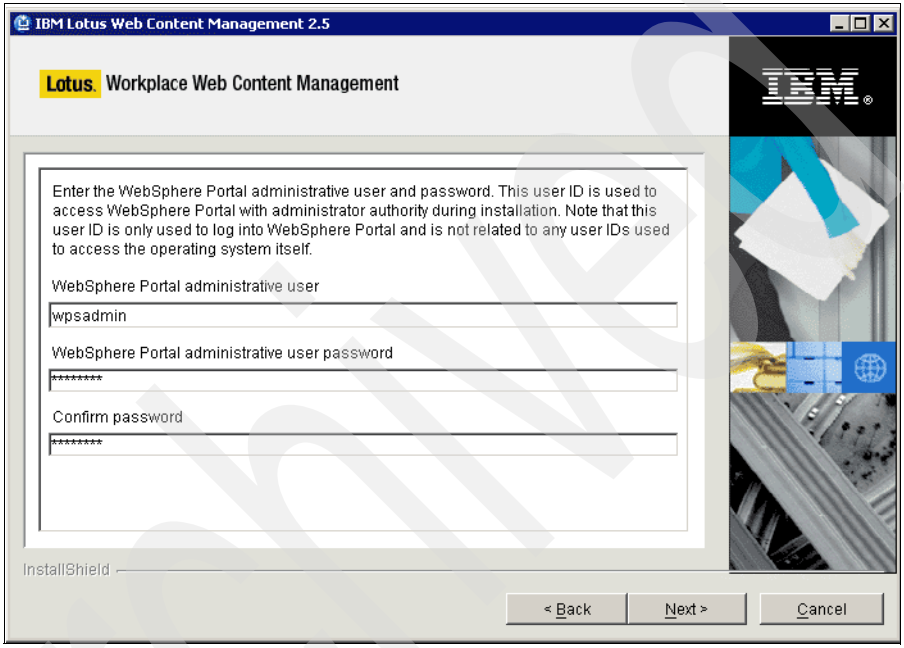


Figure 4-6 Installation WebSphere Portal administrator details

Archived

When you install IBM Workplace Web Content Management 2.5, you have the option of letting the installation process create the authoring environment for you (Figure 4-7). If you choose not to select this option you can configure the authoring environment manually.

9. Select **Yes** and click **Next**.

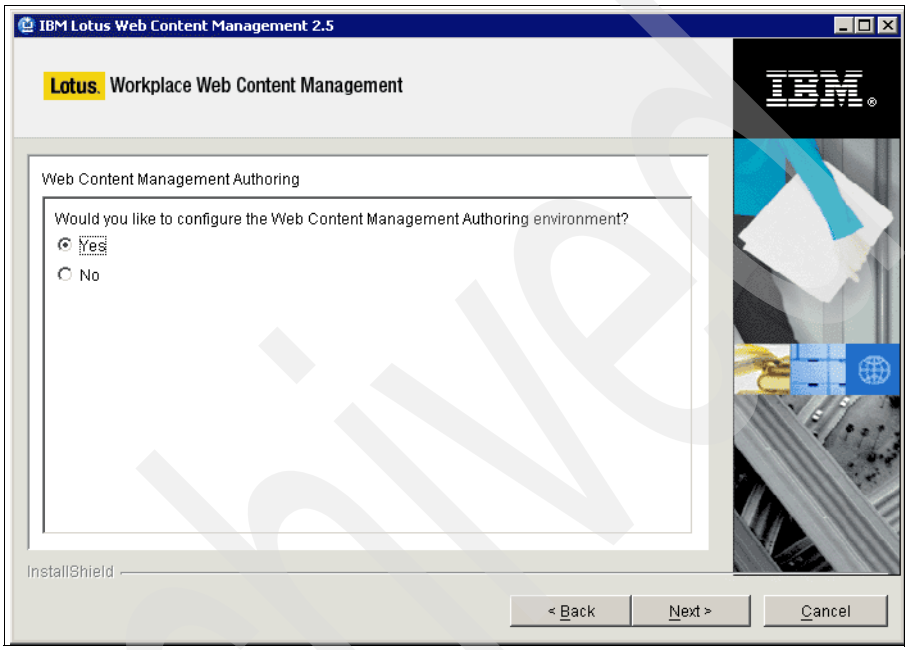


Figure 4-7 Installation authoring setup options

Archived

The installation program provides a brief summary screen with the default installation location and the amount of disk space that is needed for the installation, as shown in Figure 4-8.

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

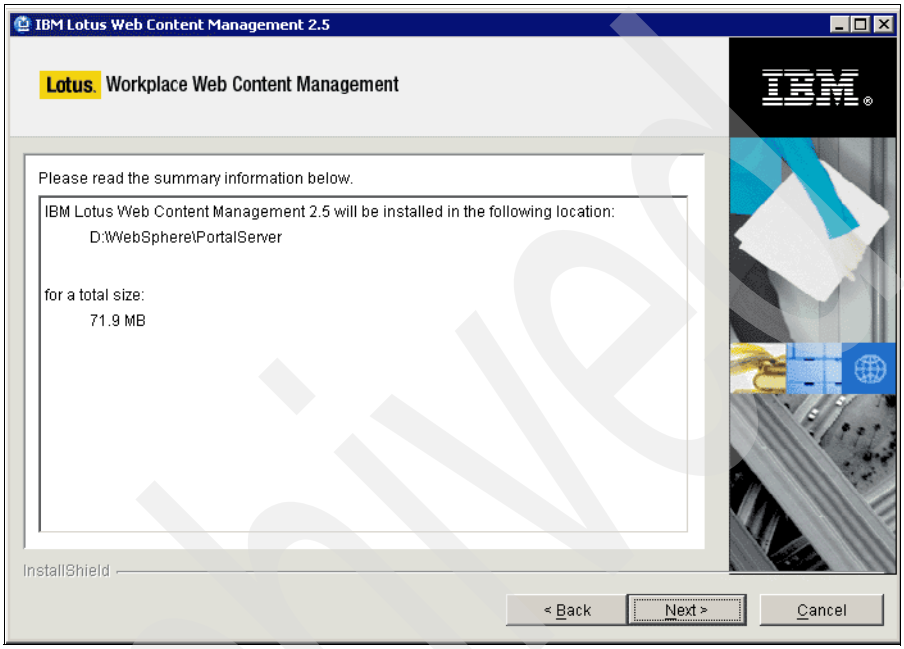


Figure 4-8 Installation summary page

When the installation completes, you see a message stating that the installation has completed successfully.

11. Click **Finish** to close the installation program.

Assuming that you let the installation program set up the authoring environment, when the installation is complete, you should be able to access your server and see a new page called *IBM Workplace Web Content Management*. To test the installation, log into your WebSphere Portal or Lotus Workplace server with a user account that is a member of the wcmadmins group. You should see a page called *IBM Workplace Web Content Management*. Clicking that link should take you to the authoring Portlet, as shown in Figure 4-9 on page 137.

Archived

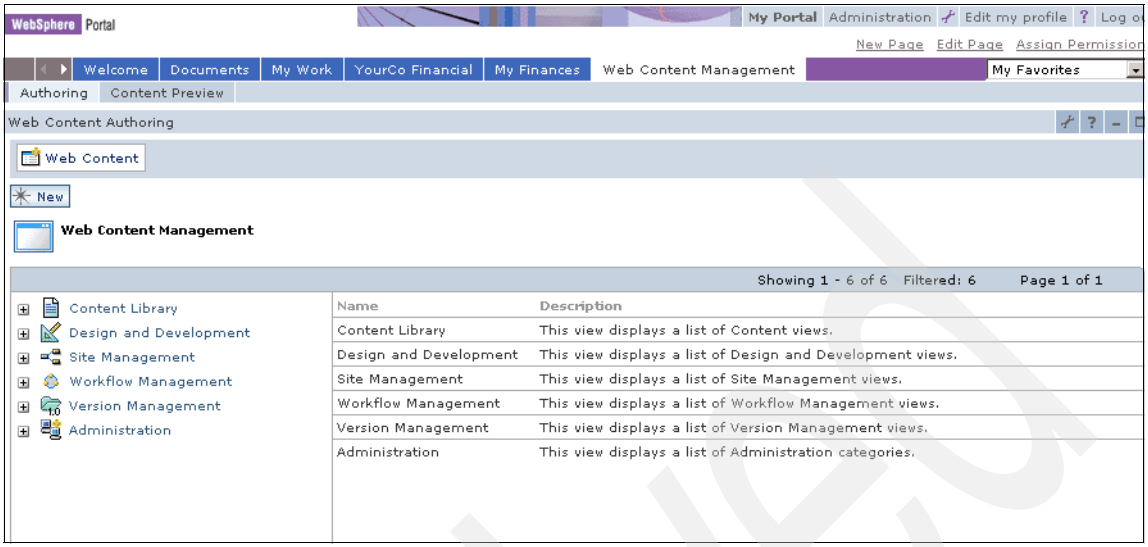


Figure 4-9 IBM Workplace Web Content Management 2.5 authoring Portlet

4.4 Configuring IBM Workplace Web Content Management 2.5

After you have installed IBM Workplace Web Content Management 2.5, you will likely want to migrate the content repository from Cloudscape to some other repository.

The process for all configuration changes in IBM Workplace Web Content Management 2.5 is identical to those for IBM Workplace Web Content Management 5.1. Refer to 3.3, “Configuring IBM Workplace Web Content Management with DB2 Universal Database” on page 77 for details about configuring IBM Workplace Web Content Management to use a DB2 server as the content repository.

For a review of all configuration options, refer to the IBM Workplace Web Content Management 2.5 Information Center at:

`http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/`

Archived

Archived

Part 3

Key concepts, information architecture, site design, and reference implementations for Web content management

Archived

© Copyright IBM Corp. 2006. All rights reserved.

Archived

5

Key concepts and terminology

This chapter addresses the core concepts of IBM Workplace Web Content Management. We review the separation of presentation and content and emphasize the importance of componentization within page design. We also describe the key items and components and recommend important considerations for reuse and extensibility.

As an example Web site, this chapter uses the River Bend Coffee and Tea Company Web site, which we refer to as the *River Bend Web site* (see Figure 5-1 on page 142). We show all concepts and terminology that use elements of this Web site. The source for the site is Lotus Education in Australia.

This chapter includes the following sections:

- Separating content and presentation
- Componentization and the page design concept
- Key IBM Workplace Web Content Management functions, items, and definitions
- Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1

© Copyright IBM Corp. 2006. All rights reserved.

Figure 5-1 The example River Bend Coffee and Tea Company Web site

Archived

5.1 Separating content and presentation

In IBM Workplace Web Content Management, the creation of content is separated from the presentation of content. This concept is a very important concept and is implemented consistently. Thus, you can create content once and then display that content with different presentations. For example, the presentation can depend on:

- The area of the Web site through which the user is navigating
- The user's personalization settings (for example, preferred color)
- Predefined presentation settings (for example, print layout)
- The online device which accesses the content

This approach also guarantees that a Web site has a consistent look and feel. If the design changes, the content parts are *unaffected*. Additionally, content creators do not need to worry about the presentation of content, which can be beneficial when the content creators do not have specific HTML knowledge. Content can be displayed using different presentations.

Archived

For example, the River Bend Web site contains a News section and a River Bend in the News section. All content documents in these sections are displayed as a list showing a title, a summary, and a link (see Figure 5-2).

Figure 5-2 River Bend in the News page

Archived

The link opens the document so that the user can read the entire text. In this case, only the title body and an image are displayed, but no summary is displayed (see Figure 5-3). These are two different presentations of the same content document and illustrate the separation of content an design.

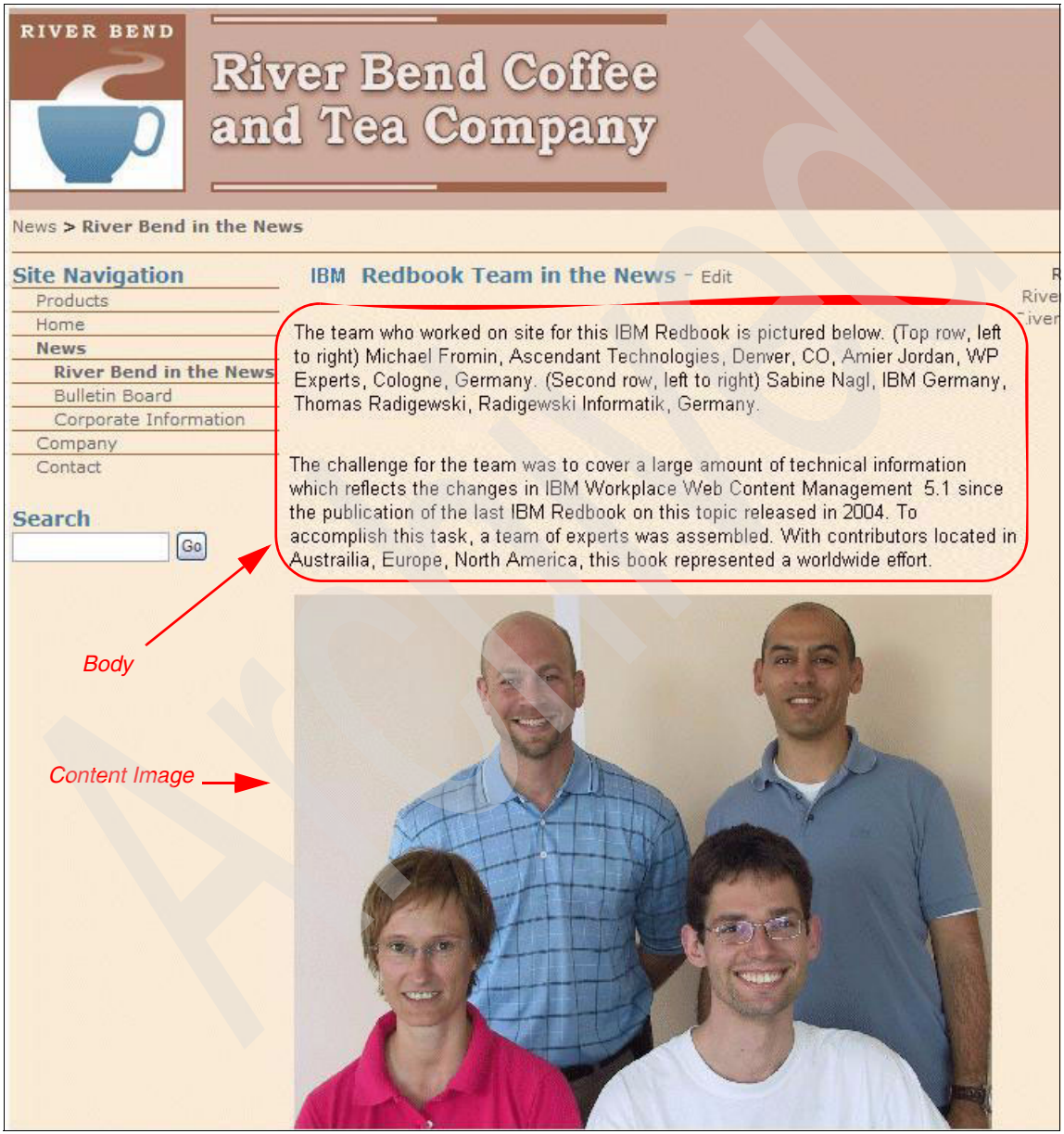


Figure 5-3 In the News document open for reading

5.1.1 Content editing considerations

Content authors frequently demand a one-click option to edit a published content page. Additionally, they request an in-place-editing and WYSIWYG editing solution. Each of these requirements are based on the same desire — the ability to edit text and content for a Web page while retaining the display format of the page as the edits are being made.

Challenge of editing separate content and presentation

Keep in mind that only some fields of a content document might be displayed on a Web site while others are not displayed. How does this impact the request for in-place WYSIWYG editing solution? Only the fields displayed on that particular Web page are available for in-place or WSIWYG editing. The other fields stay unchanged.

For example, if we apply this approach to the River Bend site, the following are true:

- A page displays the title and summary. When an author edits these fields with an in-place or WSIWYG editor, *only* these fields are available for editing.
- With the body field, no change is made unless the editor moves to a page where the contents of body field are also displayed. At this point, the editor has access to the body field. This approach requires editing the same content twice from two different access points.

You can imagine how this approach leads to inconsistency in content. Therefore, when content has to be changed, the ability to open the whole content document in an editor allows you to change fields consistently. In this way, you can achieve the separation of content and presentation.

Archived

In-place editing with the Customizable Template Portlet

IBM Workplace Web Content Management follows strictly the concept of separation between content and presentation and provides functions to add an edit-link in a Web page to open the displayed document in the Authoring Portlet or the *Customizable Template Portlet*. (For information about the Customizable Template Portlet, see 13.2, “Using the Customizable Template Portlet” on page 535.) Figure 5-4 illustrates the edit link.

Figure 5-4 Edit link for River Bend in the News page

Clicking **Edit** opens the document in the Authoring Porlet, as shown in Figure 5-5.

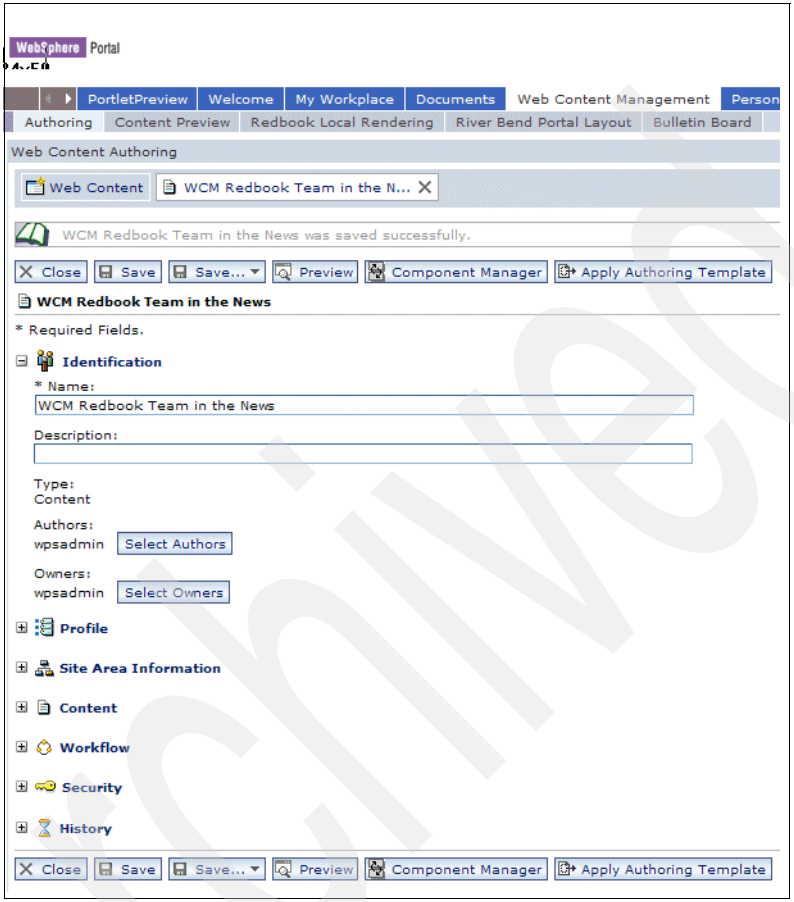


Figure 5-5 Document opened in the Authoring Portlet

Future releases of IBM Workplace Web Content Management are expected to provide enhanced editing functionality for this feature.

Archived

5.2 Componentization and the page design concept

One of the many strengths of IBM Workplace Web Content Management is the use of *componentization*. The elements of a Web page are created as *distinct units* and then are put together in a variety of ways. This function allows you to create flexible Web pages that authors can create simply and that administrators can maintain easily. Each of the elements that are created can be reused in many other pages and in other elements. So, each element should be designed from the beginning to be reusable and modular. Figure 5-6 illustrates an example Web page in which some typical component types are highlighted.

Figure 5-6 Components on the River Bend home page

Archived

Figure 5-7 illustrates the relationship between components and shows how a Web page is a composite of design and content components. In essence, a typical Web page can be constructed from the following elements:

- **Presentational layer**:

  The look and feel that controls how the page should be presented. Determining factors include:

  – What device or type of browser is intended for displaying the information

  – Font, color and other graphical components to be displayed

- **Content layer**:

  The information about the page. Determining factors include:

  – Content input using a content template

  – Tools for accessing content, such as authentication (sign in), search, and so forth

- **Information architecture layer**:

  How the content fits into the overall site information structure. Determining factors include navigational aids, such as a navigator, menus, and breadcrumbs.

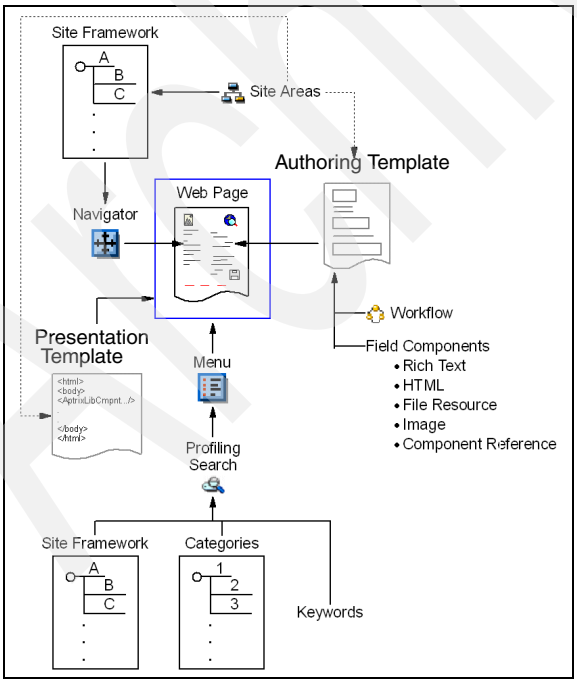


Figure 5-7 Web page component layers

The concept of componentization remains consistent throughout the different roles that are associated with creating, maintaining, and administering a Web site, namely authoring, site design, and maintenance processes. This componentization allows for the traditional, encompassing role of a Web master to be separated into a more specific role as discussed in 1.4.1, “The roles involved and their focus” on page 17.

5.2.1 Considerations for reusability

The concept of componentization is derived from the concept that a user can create a component *once* and reuse it many times in various different contexts and design areas. This reusability can be achieved by adhering to the following guidelines:

- Use a naming convention for the objects that is descriptive to make it easier for other developers to identify objects for reuse. The River Bend Web site uses prefixes such as *NAV* for navigators and *MENU* for menus.
- Identify the different types of pages in the site and look for common elements between the pages.
- Remember that you can use additional information components to allow content creators to customize a Web page to their needs.
- A site can often be broken down into a set of page styles, such as the Home page, content pages, menu pages, and so forth. You can then deploy these pages across the site to maximize reuse.

Attention: While it is true that components can be reused many times, take care to ensure that these components are not too dissected and nested such that they became cumbersome to maintain. In addition, remember that nesting components can impact page rendering and site performance.

Tips:

- Preferably, all objects should be modular so that they do not rely on any other object to close a `<table>` tag or to put `<script>` tags around it.
- You should be able to change a component slightly without having to change any other elements that rely on that component. All functionality should remain within the component that needs it (either coded directly in the component or in the form of another complete component).

5.3 Key IBM Workplace Web Content Management functions, items, and definitions

This section discusses the IBM Workplace Web Content Management items that are used to build a Web site as well as the functions that secure and maintain a Web site from a IBM Workplace Web Content Management perspective. This section summarizes functions and items and then includes an in depth discussion on their functions.

> **Note:** For a discussion that focuses on how to use these items within the context of the WebSphere Portal paradigm, namely Portal navigation, Portlet issues and so forth, see Chapter 9, “Advanced integration with WebSphere Portal” on page 403.

IBM Workplace Web Content Management defines components and items in its own technical sense. All elements in the Authoring Portlet Item Navigator which are defined within the Component Library are referred to as *components*. All other elements — from within Content Library, Presentation Templates, Authoring Templates, Site Management, and Workflow Management — are referred to as *items*.

IBM Workplace Web Content management fields

Each item is created and managed using a form with common and item specific fields. Common fields in all forms are:

- **Identification-Fields**:
  - **Name**: Required field to enter the item’s name that is used in the authoring environment.

    > **Attention:** Content items in the River Bend example Web site use a Display Title field. This field is shown on the Web site while the Name field is not. It is not required to have identical values for Name and Display Title. However, doing so eases administrative tasks.

  - **Description**: A comprehensive description. This is an optional field.
  - **Authors**: Additional authors for item. This field is not a security feature. If you need security features, you should use security options. This field is used to reference items with ID component tags.
  - **Owners**: Additional owners for item. This field is the same as the Authors field.

Archived

- Security-Fields are described in 5.3.3, “Item and component security” on page 158.
- Workflow-Fields are described in the 5.3.6, “Workflow” on page 166.

Version management

Version management is a functionality that you can use to create labeled version of content items and components. This functionality is described in detail in 5.3.1, “Version management” on page 155.

Syndication

Syndication is an architectural functionality and concept to move each instance of a IBM Workplace Web Content Management site from one server, say an authoring server to another server, such as staging or publishing server. Each instance is installed on a separate application server, and syndication ensures that several application servers contain the same data. Through syndication, content and site components can be shared seamlessly among several Web sites. The end result is identical Web sites on different servers.

Separation of the development and live environments is usual practice so that high load in the development environment does not impact high performance and speed in the live environment. Syndication of the servers in the development and live environment keeps the live server up-to-date with changes made in the development environment

Note: Syndication is different from designing a workflow. (Refer to 2.3.5, “Workflow versus syndication” on page 49 for a more detailed discussion on the comparison of this topic.)

Item and component security

Every item and component is access protected. A five-level security model (Live, Read, Edit, Workflow, and Delete) applies.

Archived

IBM Workplace Web Content Management items

For a conceptual look towards items, a separation between *content-centric* and *architectural* items might make sense. The content-centric items are used for content creation and management. The architectural items define the hierarchical structure of the Web site and its design.

Content-centric items

Creating, managing, and using content-centric items is related to the content constituency which is performed by subject matter authors and content owners. Content-centric items include the following:

- Authoring template

  The *authoring template* is used by a content creator to create content by entering text, predefined resources (such as images, files, and so forth), and components in fields.

- Categories and taxonomies

  Categories are predefined terms that are grouped into *taxonomies* that are similar to keywords, while keywords — which are used in an authoring template field — are not predefined but are text-free.

- Workflow

  Because every content item has to pass a *workflow* in order to be published, workflows are essential elements. However, their purpose is related strictly to life cycle. By default Workflow is enabled for content items but can be configured for all items and components by inserting the appropriate values in the aptrixjpe.properties file.

- Content

  Content items are any kind of information intended to be published to a Web site and are created using authoring templates.

Architectural items and components

Creating, managing, and using architectural items is related to the technical and layout constituency and is performed by IBM Workplace Web Content Management administrators and developers. For an discussion about the process and how the item values are defined, see Chapter 6, “Information architecture and site design” on page 199. Architectural items include the following:

- Site Framework: site and site areas

  The *Site Framework* defines the hierarchical site structure which is the case for the main Web site navigation.

- Design components

  A number of *design components* are available for a customized design of a Web site, such as the Navigator, Menu, and HTML components, Image and File resources, and so forth. These design components are stored in the Component Library under the Design and Development section in the authoring portlet.

- Presentation template

  In the *presentation templates*, everything comes together and the Web site can be viewed as a whole.

  Design components are referenced within the presentation template form the look of a Web page (such as the header or stylesheets), and content item fields are referenced to display text on the Web page.

Note: Now we start the in-depth discussion on the IBM Workplace Web Content Management functions and items. All examples are based on the River Bend Tea and Coffee Company Web site. If you want to build the Web site, see Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253.

5.3.1 Version management

The first intention for customers familiar with document management methodologies is likely to use version management for content and track changes during content life cycle. Although *version management* provides versioning and restore capabilities, it is not intended to replace an Enterprise Document Management system. In addition to versioning functionality for content, this function is also available for presentation and architecture layer items and components. It allows you to restore previous Web site appearances by using the aligned restore functionality.

Archived

Version management, by default, is a manual step. Content owners and content creators decide to create a version of a content item after significant changes have been made. The version is created by clicking **Save Version** in the item forms when an item is in Read mode. Figure 5-8 shows versioned content within the version library.

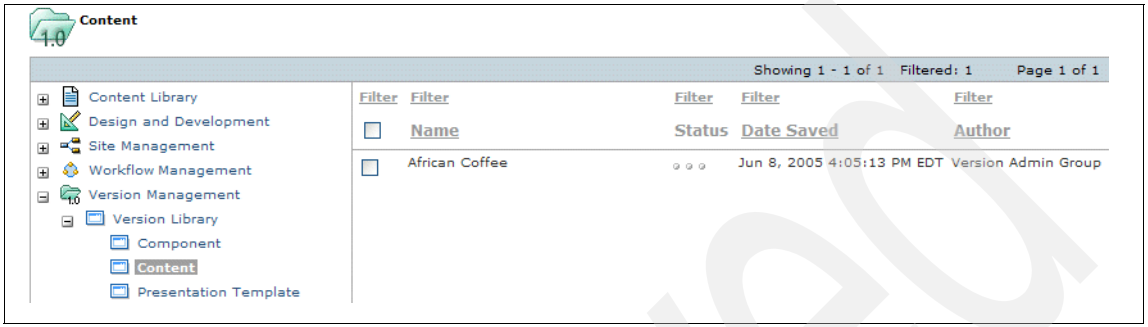


Figure 5-8 Version Library with versioned content

> **Attention:** Version management cannot be used to create a version of an IBM Workplace Web Content Management Web site, nor does it replace the need for backups.

After installation, version management is enabled for selected items and all components defined in the component library. Versonmanagement can be enabled by the IBM Workplace Web Content Management server administrator for all items. For details on how to do this, see the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

The Save Version button in an open for viewing document is available for the version enabled item. If you use the button, a label has to be specified.

> **Attention:** The label is a descriptive element for a version. Version numbering is incremental and cannot be changed.

To view all versions of an item, click the item link. All versions are displayed, as shown in Figure 5-9 on page 157.

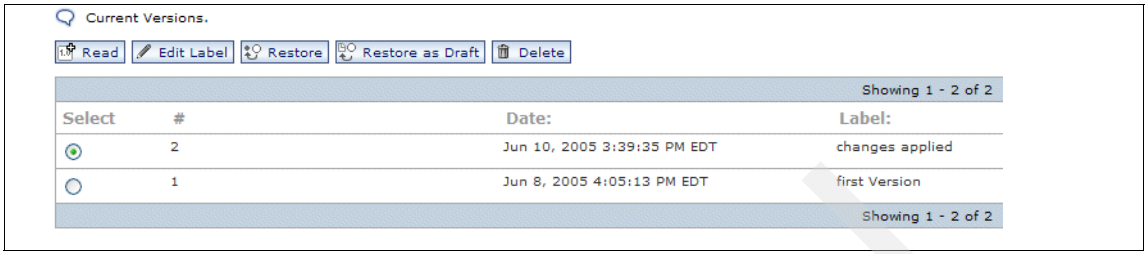


Figure 5-9 Version library shows all versions and labels of a versioned item

A version-all function is available in the version library. This function creates a new version of all version-enabled and non-workflow enabled items and components in the version library. This function provides a snapshot of all existing items but not of a Web site. The version function is not aligned with consistency checking. Missing components in references or relations on a Web page are not identified.

A workflow-enabled item is only versioned in the published status. A restored workflow-enabled item matches the status and position of the original item. When versions are created they can be restored separately or in a reverse function to the Version-all function. Therefore, a batch restore function is available in the version library. Versions to restore are identified by date or label.

To use batch restore identical labels might be a requirement. The label all function — also available in the version library — allows you to label all recent versions for easy identification for a batch restore.

5.3.2 Syndication

Syndication is the method used by an IBM Workplace Web Content Management server to replicate data from one IBM Workplace Web Content Management server to another. Unlike Workflow, Syndication is not involved in the process of approving content. Syndication is only responsible for distributing IBM Workplace Web Content Management assets across multiple servers.

Two items are required for Syndication:

- The *Syndicator* contains the data that is copied to another application.
- The *Subscriber* retrieves the data copies from the Syndicator.

Syndicators can syndicate to multiple Subscribers, and Subscribers can receive data from multiple Syndicators. When creating a Syndicator, IBM Workplace Web Content Management allows you two choices for the granularity of items to be syndicated:

- All items

  When configured with this setting, all elements of the IBM Workplace Web Content Management server are replicated to the other server, including draft and expired content. Typically, authoring and development servers need access to all types of content for testing purposes. This setting is common when syndicating between distributed authoring servers or between an authoring server and development server.

- All live items

  All live items replicates only live content. Live content is defined as content that is approved via a Workflow (live) but is not yet past its expiration date. Typically, a production server would only receive live data. So, this setting is common when syndicating from authoring to staging or staging to production.

Tip: Syndication always syndicates copies of technical assets (for example, presentation templates, authoring templates, library components, and so forth). By default there is no setting to stop this syndication. You can configure IBM Workplace Web Content Management to Workflow these items, at which point they would follow the same rules as content.

5.3.3 Item and component security

IBM Workplace Web Content Management offers comprehensive object-level security with its own security scheme. In each item or component, four security levels are available, plus a Workflow security if the item is in a Workflow. Without the appropriate security level assigned, users are not able to perform tasks or to see the content or design elements in a Web site or portlet.

Users and groups can be assigned the following access rights, listed from lowest to highest access level:

- Live

  Read-only mode for published content.

- Read

  Live access plus read-only mode to objects via the user interface.

- Edit

  Read access plus create or edit mode to objects via the user interface.

- Workflow

  Approve or edit access during a workflow approval stage. This level is only assigned to users while an item is in the approval stage of a workflow.

- Delete

  Edit access plus delete mode to objects via the user interface.

Access rights are assigned in the item or component interface in the Workflow section, as shown in Figure 5-10.

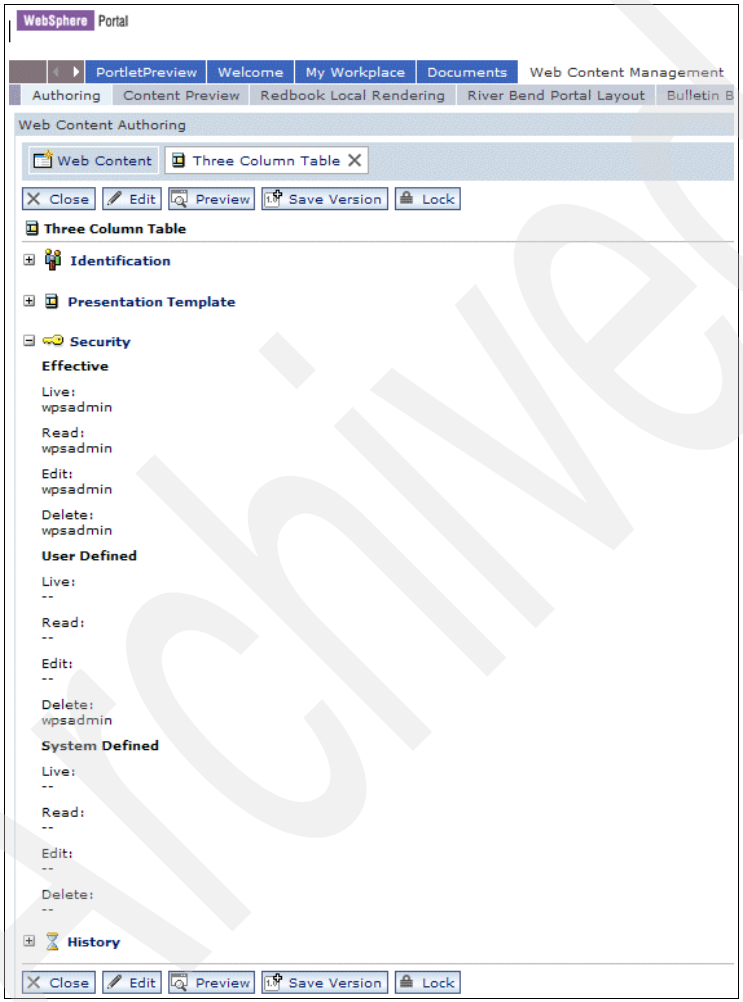


Figure 5-10 Security section in item

By default, a *system-defined* security level and a *user-defined* security level are assigned to every object. The system-defined security level can only be set by the site administrators. The user-defined security level allows a user to set access rights to an object up to and including the level at which the user has that

Archived

right. User-defined and system-defined security settings combine to give an object its *effective security*.

Workflow security

All content must be assigned to a Workflow approval process. During a Workflow, Workflow security is applied automatically, as defined in the Workflow stages, and is given the appropriate users access to the item. In addition, a *workflow-level* security level can be applied to other objects. Content to enhance and to refine the security settings, by inserting the appropriate values in the aptrixjpe.properties file, is shown in the following:

```
# control properties
control.Style=com.aptrix.pluto.workflow.WorkflowControl
```

In Workflow stages, a Workflow security section is visible and displays security settings, as shown in Figure 5-11.

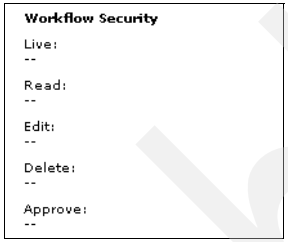


Figure 5-11 Security section showing Workflow security

For example, a draft content item might have a specific security level that is assigned by the Workflow stage and that is set such that only the authors have access to *edit* the content. When the content moves to the approval stage, only a group of approvers have edit rights to the content, while the author's access is changed to read. Then, the content moves to the next Workflow stage of *published*. The published Workflow stage changes the security level to *live* for anonymous users and to *edit* for the content author.

Workflow security replaces user-defined security for an item that participates in the workflow-level security. Both workflow or user-defined and system-defined security settings combine to give an object its *effective security*.

Interestingly, a Workflow stage is also an *item* and, therefore, is subjected to both item and Workflow security. (It is not recommended, however, to control and modify security for a Workflow stage through Workflow security). For more information, refer to the Information Center available with your product download.

Archived

Conclusion

IBM Workplace Web Content Management security is constructed by the following layers:

- Portal security

  Provides access to the Portal server and applications. For Web site readers, access to the Portal environment — especially portlets that display content — is sufficient. In addition, for all other users, appropriate access to the other layers has to be granted. Instructions about how to grant access to the Portal are found in the Portal Information Center or in *IBM WebSphere Portal for Multiplatforms V5.1 Handbook*, SG24-6689, available at:

  http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg246689.html?Open

- Portlet security

  In order to perform tasks in IBM Workplace Web Content Management — read, edit, approve, and manage items — access to the authoring portlet has to be granted. For instructions about how to configure the authoring portlet access, see 3.5, “Installing the Rendering Portlets” on page 95. For instructions about configuring the Customizable Template Portlet, see 13.2, “Using the Customizable Template Portlet” on page 535.

- IBM Workplace Web Content Management item and component security

  Defines final access to each individual item and component in the authoring portlet and the Web site, either stand-alone or in a portlet. If security is not set appropriately, the item cannot be viewed, edited, approved, or managed.

5.3.4 Authoring template

Authoring templates are used for content creation. They provide the editing and authoring forms for content authors. Similar to document types in document management solutions, fields for text and metadata are defined in the authoring template. In addition to the common fields that are described in “IBM Workplace Web Content management fields” on page 152, the following sections are provided:

- Profile
- Content

These sections contain fields for immediate content input or references to other components.

Profile

The Profile section contains a Categories and a Keywords field. Both fields can be regarded as metadata to characterize a content item, as shown in

Figure 5-12. Categories are selected from the elements that are defined in the Category Management (see 5.3.5, “Category Management” on page 164), meaning that they are *predefined*. The advantages of Categories is that they provide quick selection with no misspellings or duplicates but slightly different elements. The disadvantage of Categories is that only predefined Categories are available.

The Keywords field allows users to enter any text without restriction. Keyword fields are not selected from a predefined list. The advantage is that the user can enter any word or phrase. The disadvantage is that you cannot check for misspellings or duplicate entries with only slightly different meanings.

Categories and Keywords are used as content metadata for search and Menu definitions. See “Menu component” on page 182 for more details about how to use Categories and Keywords in menus.

Figure 5-12 Categories and Keywords fields in Profile section of Authoring Template

Content

The Content section lets the authoring template reflect the companies’s content types. All fields are added to the authoring template using the component manager, as shown in Figure 5-13.

Figure 5-13 Component manager

The fields are defined by a field name and the field type (as shown in Figure 5-14). The field type is selected from a list that is either a component field, such as a text, a Rich Text field, or a reference of a component that is defined in the component library (see “Component library” on page 177). Thus, the field type reuses a predefined or predeveloped component.

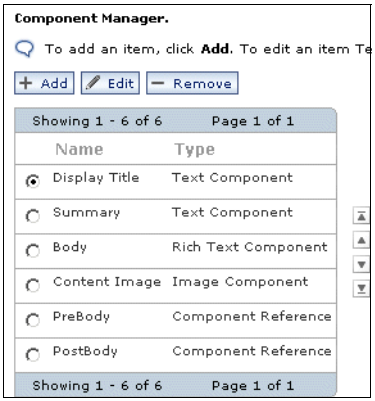


Figure 5-14 Fields in component manager

For example, the River Bend development authoring template uses the following (as shown in Figure 5-15 on page 164):

- The Text field for the display title and summary.
- The Rich Text field for the body.
- An Image Component for an image.
- The Postbody and Prebody fields for Component References where the reference to a Menu is entered.

Archived

Figure 5-15 Riverbend Authoring Template

5.3.5 Category Management

The Category Management section of the item navigator defines taxonomies and categories. These items are an important, and often underestimated, elements for content that is metadata.

Note: Categories are metadata for content that is used in menus and search.

Taxonomy

A *taxonomy* defines the collection of terms that form the categories. For distinct term-sets, you can create independent taxonomies, although it might not be necessary because the number and depth of categories is nearly unlimited. As authors select categories, they assign them to a content item by clicking a tree of categories. A reasonable depth of the taxonomy tree helps the user in selecting categories. A maximum of five to six levels has proven to be a good depth with which to work. You can adjust taxonomies where necessary.

Category

A *category* is an informational component that is used to describe what a piece of content is about. Categories used in content transcend hierarchies of architectural elements as site or site areas. Changing the navigaitonal hierarchy is difficult, while categories should reflect business needs and are adjusted as the organization's needs change.

Do not confuse categories with a *site area*, which is created to provide a path to that piece of content. Menu design components can use a category, via search criteria, to create menus or to create lists of related documents. For example, if the coffee business faces a boost for Ginger flavored Coffee, the River Bend company can add the category *ginger* to the taxonomy. The overall hierarchy based on site and site areas stays unchanged. Immediately after adding the category *ginger* to content documents, searches for *ginger* on the River Bend Web site show results without any development work. In addition, the category *ginger* can be removed easily without any influence on the overall navigational hierarchy of the site.

Another example is that the River Bend Web site uses categories in a menu to identify content of a specific topic. A products page displays coffee from Colombia, while another page displays Espresso coffee. Now, a coffee bean from Colombia is also a good Espresso coffee bean. Adding the category *Espresso* to the Rich Colombia Coffee document makes the category appear automatically not only on the Colombia page but also on the Espresso page without any additional work (Figure 5-16 on page 166).

Archived

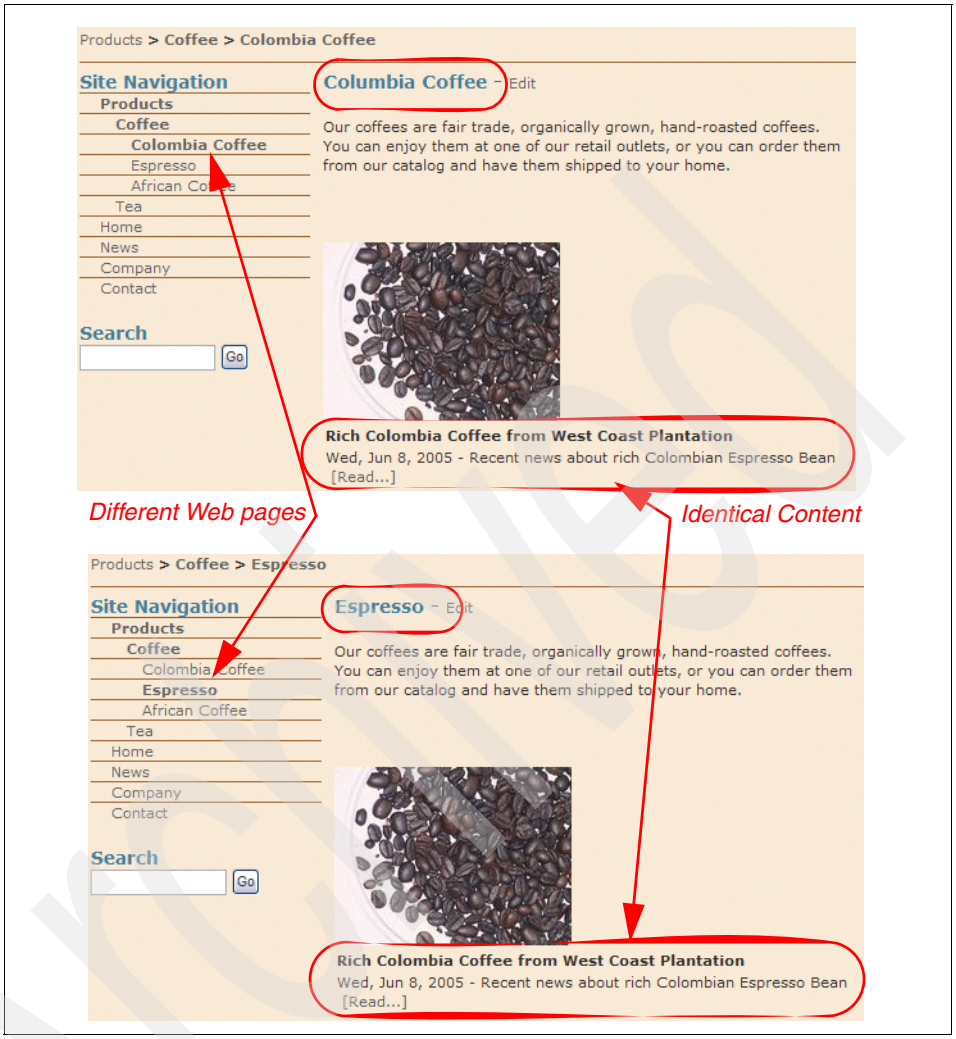


Figure 5-16 Additional category Espresso makes a document appear on two pages

5.3.6 Workflow

A *Workflow* comprises one or more stages. Because Workflows are used mainly for content, this section discusses how to use a Workflow that way. The Workflow is targeted to move content through its life cycle. Appropriate *stages* can be created to support the life cycle. The stages can or cannot contain *actions* that make the content move into the next stage or that just send e-mail messages.

Workflow stages are conducted one after the other. There is no parallel processing. However, approval from several approvers can be part of a Workflow using the joint approval option. During the approval stage, all approvers have to approve an item before it is pushed to the next step. Otherwise, the item is rejected.

Figure 5-17 illustrates the basic Workflow dialog box, indicating the document's status and the current workflow stage. In addition to the Publish and Expiry date fields, two more date fields are available for individual purposes (API interaction). Workflows do not focus on anything but life cycle management. There is no such thing as transactional workflow management, automation, or escalation management.

To achieve such challenges a dedicated Workflow product has to be used, such as IBM Process Choreographer. You must use the API to integrate.

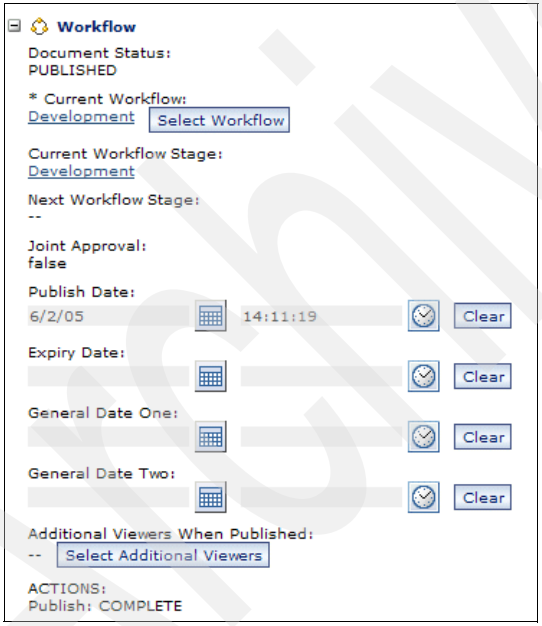


Figure 5-17 Workflow section

Workflow stages

Before a piece of content appears on a Web site, it must pass through all Workflow stages that are assigned to it. At each stage, security can change and can allow or prevent a person to create, update, reject, or publish content. A typical sequence of stages for an approval Workflow, which at the end sends the

Archived

content to an expire stage, could contain the following stages, as shown in Figure 5-18:

- Draft
- Approval
- Reject usually sends the item back to the Draft Stage
- Publish
- Expire

Figure 5-18 Workflow Stages for an approval Workflow

If content is rejected at any stage, the content author must correct or amend the content and resubmit it for approval. A document can be sent to a specific stage following the rejection. The reject stage is set up per Workflow.

A part from the approval Workflow shown in Figure 5-18, the most basic Workflow that can be created moves a Web page directly from *draft* to the *published* stage. For example, the River Bend Web site uses a complete approval Workflow that is named *Approval Workflow*. It contains the draft, approval, reject, publish, and expire stages. (See 7.7, “Building a workflow” on page 268 for more information.)

Workflow actions

Each stage can contain *actions* that are executed when entering or leaving a stage. Available actions by default are:

- Publish
- Expire

You can create the following additional actions:

- E-mail
- Scheduled Move

Typically, the *publish action* is executed when entering the publish stage, or it is triggered by the value of the publish date field in the content’s Workflow section. Thus, approved content appears on a Web site only on the published date. The

Archived

same is true for expiring content with an expire action that executes on the expire date that is defined in the item form.

A *scheduled move action* can move content from the publish to the expired stage and have the action disappear from the live Web site. This action keeps the live site up-to-date and reduces human interaction.

Among the actions that can be created as desired is the e-mail notification. The example in 7.7, “Building a workflow” on page 268 uses the e-mail action to inform users about the rejection of content. The e-mail action can also be used to inform approvers about content that is waiting for approval.

5.3.7 Content

Now, we have discussed all the elements that you need to create content: Authoring template, taxonomy and category, and workflow. Creating content for users who rarely use the authoring interface might be confusing. Apply security to all items to make sure that the authoring portlet shows only items that users need to see.

Optionally, the Customizable Template Portlet is a good choice because it allows you to hide fields that users do not need. The Customizable Template Portlet is very easy to use. Fields are preselected, such as the authoring template and site area. Thus, one Customizable Template Portlet is for one type of content only, unless you do more customization. (See 13.2, “Using the Customizable Template Portlet” on page 535 for more information.)

Creating content using the authoring portlet

After selecting the appropriate authoring template for the intended content, you have to select the site area to which the content is linked. The site area is the hierarchal location for the content. Navigators use this hierarchy to display the content that is linked to the different site areas. Web site visitors find the content when navigating through the Web site. After the content is saved, you can link it to other site areas to create a broader usage of the content item.

For example, the River Bend Web site offers coffee from different countries. Information about a new harvest of Colombian coffee should appear in the Columbia page. However, because Colombian coffee is sold in shops in multiple locations, the shop information pages for different locations should also contain that document. So, in addition to being linked to the Colombia site area, the content is also linked to other locations. Then, the fields are filled with the content. If authors tend to be creative with their content, you might want to give them the opportunity to express the creativity in Rich Text and image fields. On the other hand, limiting creativity by providing simple text fields might be the appropriate choice because all design elements are used in the presentation

layer to apply corporate design, including fonts and font sizes, colors, and alignment. The same principle applies to using HTML text or PHP.

Many users prefer to use pictures in their Web content. Predefined image resources provide images that fit the design in size and color, which can be used through component resource fields. However, this use can limit the creativity of content authors to a reasonable level. You should define the appropriate set of options that are available to content creators and apply those options to the authoring templates to fit corporate needs and objectives for using IBM Workplace Web Content Management.

5.3.8 Site framework

A *site framework* reflects part of the information architecture of a Web site. A lack of investment in the information architecture can result in an accepted Web site that has a poorly architected site framework. Unfortunately, this situation is often identified *after* IBM Workplace Web Content Management projects have already progressed into late stages. For a discussion how to achieve a meaningful, comprehensive and well-accepted site framework, see Chapter 6, “Information architecture and site design” on page 199.

A site framework provides a hierarchical structure for the Web site and is the base for the final Web site’s navigation. The content is linked to the site framework and is identified for display on the Web site using the content’s site framework information as search criteria for navigators and menus.

The navigator and menu design components use the site framework to create a site map (also a navigator), navigation, breadcrumbs, and menus. Because these elements are created dynamically at run time, there are no broken links when a site area is changed or moved.

Archived

Figure 5-19 shows River Bend's site framework and the resulting site map navigator.

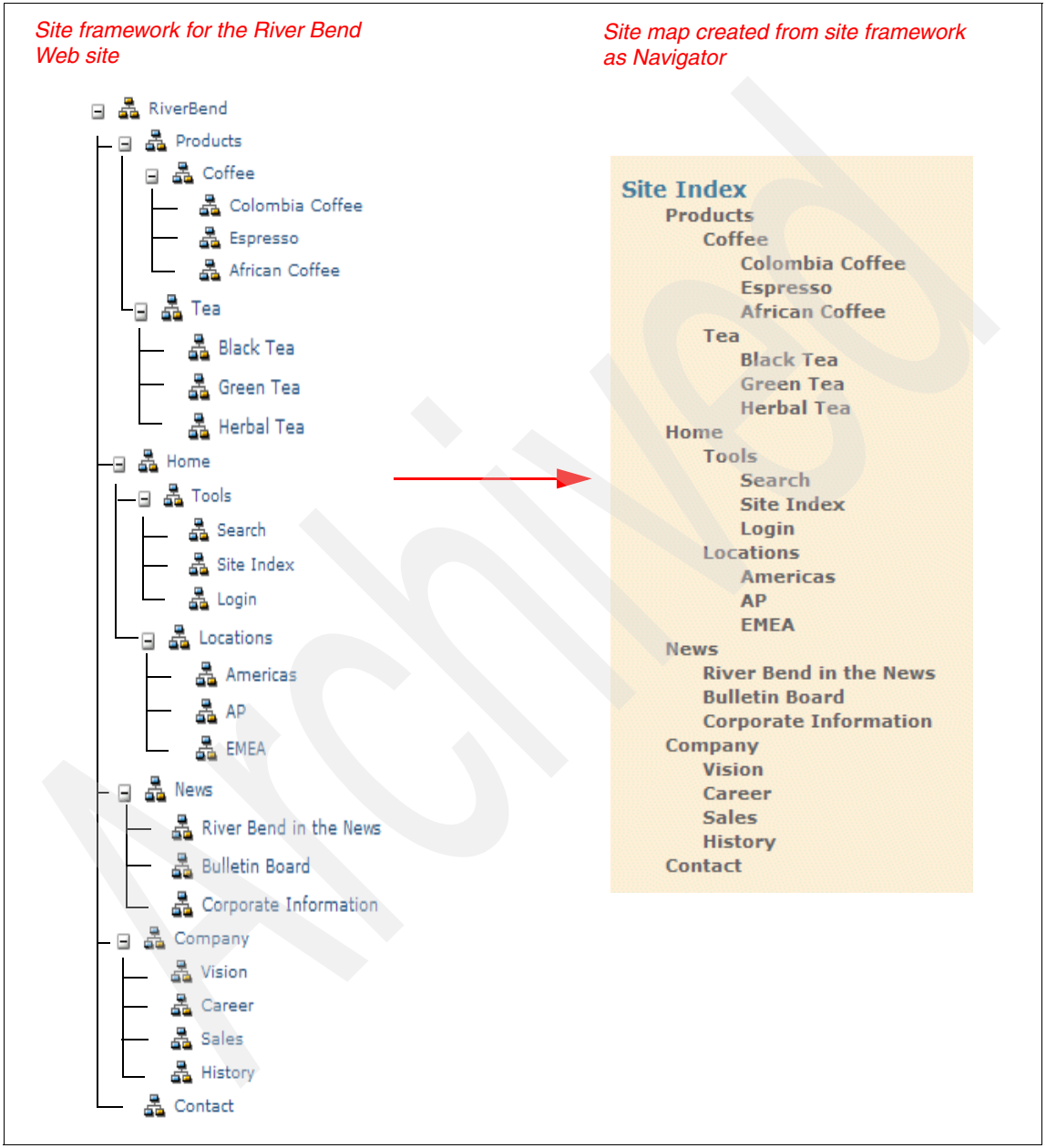


Figure 5-19 River Bend site framework and site map

Archived

Site

A *site* is the root for Web site's hierarchical structure. Although it is possible to manage several sites in the IBM Workplace Web Content Management site management interface (for example, several Web sites are hosted), one Web site is based on one site only. The site can define relationships between authoring templates and presentation templates for the whole site as shown in Figure 5-20. These settings are overruled by definitions set in site areas.

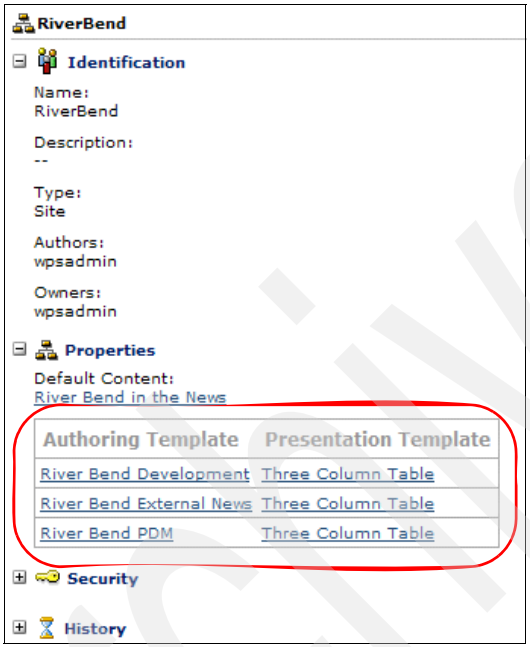


Figure 5-20 River Bend site that pairs authoring templates and presentation templates

Site area

A *site area* is a unit within a site framework where content items are grouped together. You have to define the default content for a site area. The default content displays when the page is opened. Without default content assigned, the page cannot be viewed in the browser.

Default content, of course, can contain component references that allow individual designs for different site areas and content pages. In the River Bend Web site, some content items reference a menu component. The menu displays content that is found using the menu's search criteria. (See "Menu component" on page 182 for details about menus.)

Archived

Before the content can be viewed on a Web site, each authoring template has to be paired with a presentation template. Doing so on the site level underlying site areas can inherit the pairing. Pairings on site area level can define new pairs for the specific site area and below.

Site areas allow the use the component manager to add component fields or component reference fields to the site area document. (Identical functionality is available in content items.) Using this feature in site areas allows you to define design or content items to be displayed on selected site areas.

Creating and maintaining site areas is performed by site administrators rather than by users or content creators. Site administrators are expected to be more literate and advanced using components. Therefore, it might be easier for them to manipulate the design of site areas than placing the task into content creators' hands who need extra guidance to understand the component library philosophy.

Referencing a menu in a site area rather than in a content item (as shown in 7.19.1, "Referencing site area menu in specified content pages" on page 316) has the advantage that the menu item is predefined in the site, and an content author does not have to know how to do perform the task or which component to reference.

Site and site areas in a site framework can be classified into parents, siblings, and children (Figure 5-21), where the site is the parent for preceding site areas but has no parent or sibling itself because it is the only root for a site framework. Site areas always have a parent and often siblings and children.

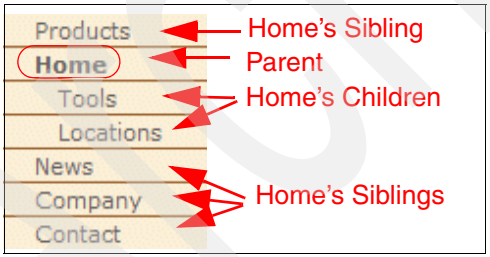


Figure 5-21 Parent, sibling, and children hierarchy

Archived

5.3.9 Design and development components

Having discussed all content related and main architectural elements, we now discuss the design and development components. Design and development components define the Web site's appearance and assure corporate identity.

Presentation templates

In the presentation template, everything that we have discussed thus far and all the components come together to form the Web pages that are viewed by site visitors. The presentation template combines all layout and design elements needed for the Web site appearance, as shown in Figure 5-22.

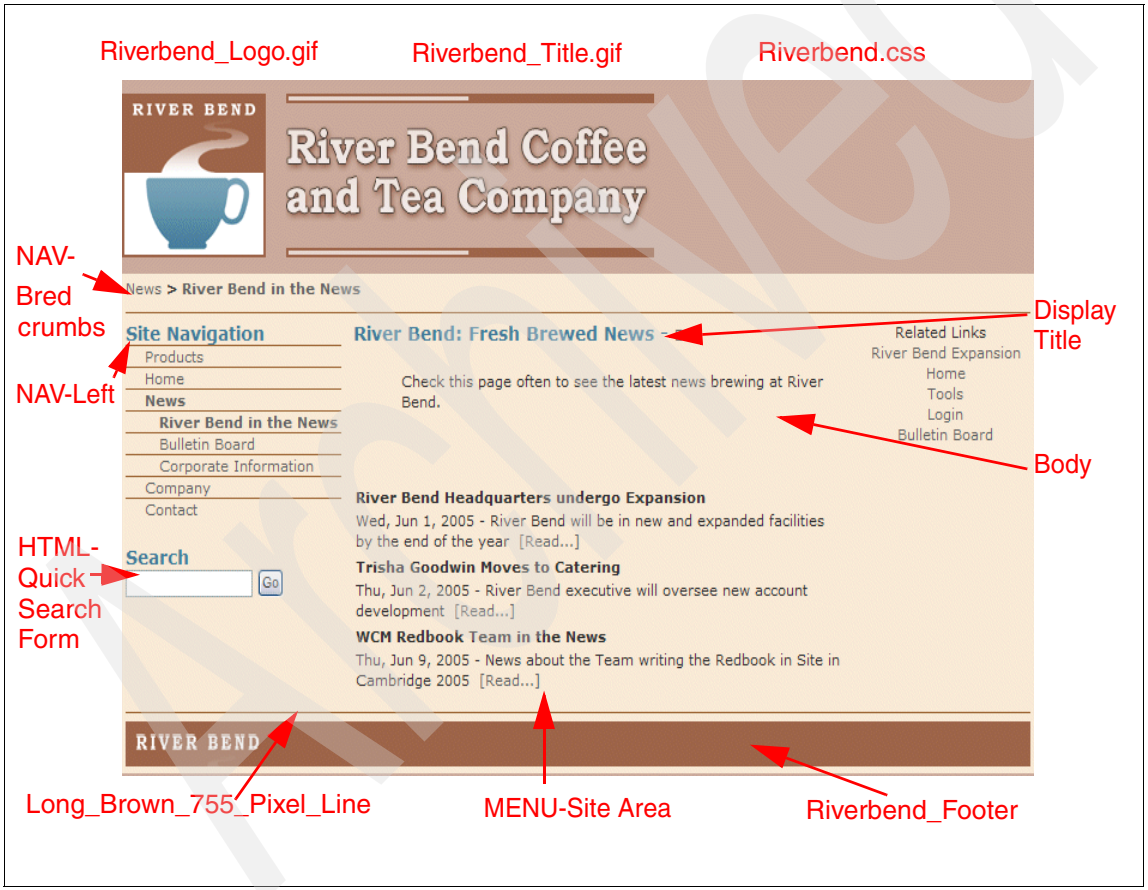


Figure 5-22 Site area for River Bend in the News and its referenced components and items

The River Bend Three Column Table presentation template displays a stylesheet and several images to create the overall layout which is consistent for all pages.

IBM Workplace Web Content Management references components in the component library by its name, as shown in the following example:

```
<link rel="stylesheet" href='<AptrixLibCmpnt name="Riverbend.css"/>'>
<AptrixLibCmpnt name="Riverbend_Logo"/>
<AptrixLibCmpnt name="Riverbend_Title"/>
<AptrixLibCmpnt name="Long_Brown_755_Pixel_Line"/>
<AptrixLibCmpnt name="Riverbend_Footer"/>
```

A site navigator is referenced at the left side of the page as well as a breadcrumbs navigator beyond the logo area (Figure 5-23).

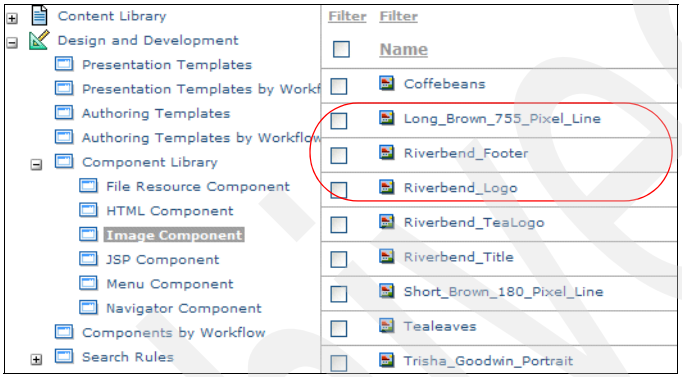


Figure 5-23 Image components

These are library components that are referenced in the same way as the image components, as shown in the following example:

```
<AptrixLibCmpnt name="NAV-Left"/>
<AptrixLibCmpnt name="NAV-Breadcrumbs"/>
```

The navigator a search area displays using an HTML component. The HTML component is also a component library component and is referenced as the other components by its name, as shown in the following example:

```
<AptrixLibCmpnt name="HTML-Quick Search Form"/>
```

The component library components state clearly that content does not play a role in the layout of a page.

The content is referenced in the presentation template. In the center area, content items fields are referenced. Content items are stored in the content library that is sorted by several different criterias. The Three Column Table presentation template references the Display Title, Prebody, Body, Content Image, and PostBody fields using the field names as identifiers or keys.

Archived

All fields are referenced in a current context. The current content is the content item to which the user has navigate using the navigator on the left side of the page. The type content states that a content item is referenced, as shown in the following example:

```
<AptrixCmpnt context="current" key="Display Title" type="content"/>
<AptrixCmpnt context="current" key="PreBody" type="content"/>
<AptrixCmpnt context="current" key="Body" type="content"/>
<AptrixCmpnt context="current" key="Content Image" type="content"/>
<AptrixCmpnt context="current" key="PostBody" type="content"/>
```

The PostBody does not display a simple text value such as the Display Title field but displays a component. For the River Bend in the News content page, the component reference MENU-Site Area is entered in the PostBody field, as shown in Figure 5-24.

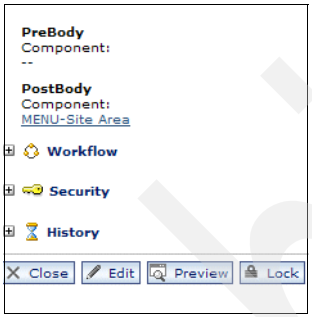


Figure 5-24 River Bend in the News PostBody Component

The menu displays all content items that are found in the site area for River Bend in the News. As shown in Figure 5-22 on page 174, there are three content items displayed in the menu.

Alternatively, a component from a site or site area could be referenced. The River Bend Web site uses this alternative for the Colombia Coffee, Espresso, and African Coffee site areas. These site areas contain a component reference which references the MENU-Site Area. To display the menu in the presentation template, the following reference with the type `sitearea` is used:

```
<AptrixCmpnt context="current" key="Menu" type="sitearea"/>
```

For the River Bend Web site, the effect on the page is identical to the use of a menu in the PostBody field. However, the reference in the site area is created by a site administrator who is familiar with the existing components and their usage, where content authors are not trained using components but are familiar with how to create content.

Thus, referencing components from within a site or site area can reduce training needs for users and can increase design flexibility.

Note: If a referenced component does not have a value to display, nothing is shown on the Web site. The PreBody and Content Image fields in the content item that are displayed in Figure 5-25 on page 178 are empty, and the references do not appear on the page.

Note: When a new presentation template is created, remember to pair it with the appropriate authoring template in the site framework.

Component library

The component library is the home of all components. Each of them can be referenced with the component library tag throughout the other items and also in the components themselves.

Navigator component

The navigator provides a mechanism that allows users to move around the site. A navigator uses the site framework to create links to different site areas or documents in the site automatically. Because it is based on the site framework, the navigator displays the hierarchy of the information architecture. Thus, when a site framework or a site area changes, the navigator is updated dynamically, eliminating the need for re-coding and preventing the possibility of broken URL links.

For example, the River Bend Web site displays a navigator on the left side, called NAV-Left, which offers the main navigation through the entire site framework, making all site areas available for browsing. This navigator displays site areas beyond the root, which is the site named River Bend. Browsing through the navigator and clicking a navigator section or site area displays one (or more) children site area levels (if children are available). The sections that a user selects or activates for browsing are highlighted in bold, indented, and separated by a small brown line. For each level (that is, each site area), the default content for the site area is shown in the middle of the page, as shown in Figure 5-25 on page 178.

Archived

Figure 5-25 NAV-Left on page

When building the navigation, you have to define the navigator start site area. For the NAV-Left, the start area is selected and the River Bend site is chosen. Thus, navigation starts from the root of the site framework. The navigator does not include the node start, because the root node for the River Bend site is not displayed in the navigator.

Because the River Bend Web site does not have ancestors, the ancestor level is none. The navigator displays one descendant level when a site area in the navigator is activated. Also, note that the preceding sibling's level and next sibling's level are not relevant because the navigator starts already at the root level without including a start node, which is a requirement for the sibling values. If we start the navigator at a lower level, we would have a reason to include siblings for navigation. In the case of the River Bend site, however, sibling values do not exist.

Archived

Figure 5-26 and Figure 5-27 explain the ancestor, sibling, and descendant level concept. In Figure 5-26, *Home* is the selected site area. A preceding sibling is *Products*, and the next siblings are *News*, *Company*, and *Contact*. No ancestor is shown. The first level descendants for *Home* are *Tools* and *Locations*.

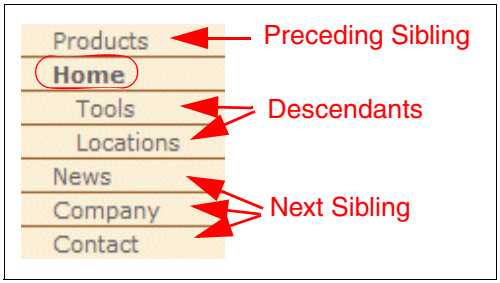


Figure 5-26 Descendants and siblings in the RIver Bend navigator

In Figure 5-27, *Tools* is the selected site area. *Home* is the ancestor. The descendants for *Tools* are *Search*, *Site Index*, and *Login*. The next sibling is *Locations*, and there is no preceding sibling.

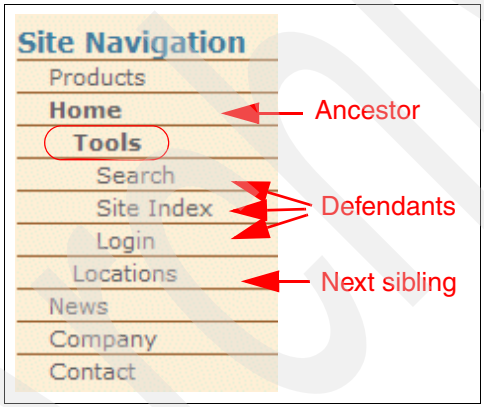


Figure 5-27 Siblings and children in the River Bend navigator

Archived

The definitions and values for NAV-Left are as follows (illustrated in Figure 5-28):

- *Show Site* displays the River Bend site in the navigator, which is not intended. Remember, *Include Start* is not selected for the selected start area, so *Show Site* reverses that choice.
- *Show Content* is not selected, so no content names are seen in the navigator.
- Selecting *Expand Current* expands the site area and shows one descendant level.
- The option to highlight navigator content using component designs is not selected. You can format for active navigator sections using code in the Header, Footer, Separator, and Component Design fields.

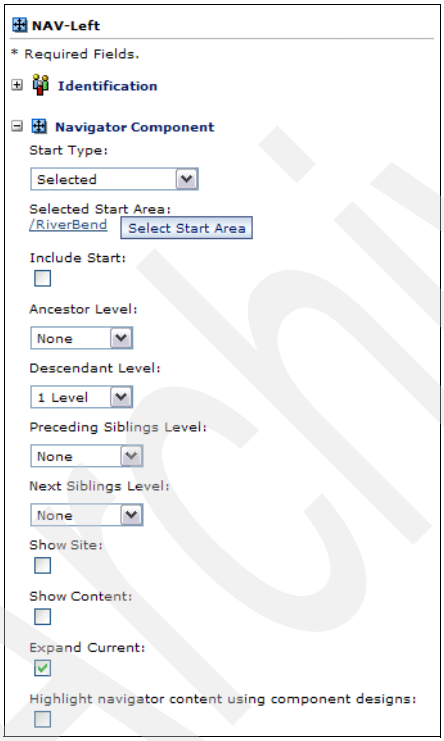


Figure 5-28 Definitions and values for NAV-Left

The Header text, *Site Navigation*, is formatted with the contrastHeading font class as shown in the following example:

```
<font class="contrastHeading">
Site Navigation
</font>
```

Archived

The footer stays empty and the Short_Brown_180_Pixel_Line image resource is referenced as a separator for navigator sections as shown in the following example:

```
<AptrixLibCmpnt name="Short_Brown_180_Pixel_Line"/><br>
```

Component Design 1 defines the indent of three spaces, and the alternate design — plain text versus bold — is defined using HTML components for the navigator sections as shown in the following example:

```
<IndentCmpnt offset="0" repeat="   "/>
<AlternateDesign highlight="HTML-Layout-Main Nav Bold"
normal="HTML-Layout-Main Nav Plain" type="Any"/>
```

This is one example of how a navigator can display site navigation. River Bend Web site uses other navigators in addition to the NAV-Left navigator.

The navigator *NAV-Site Map* displays all site areas at once (Figure 5-29). The displayed sections or site areas are reference links to the default content for the activated site area. The references for links in the NAV-Site Map uses placeholder tags as follows:

```
<a href='<Placeholder tag="href"/>'><b><Placeholder tag="name"/></b></a><
```

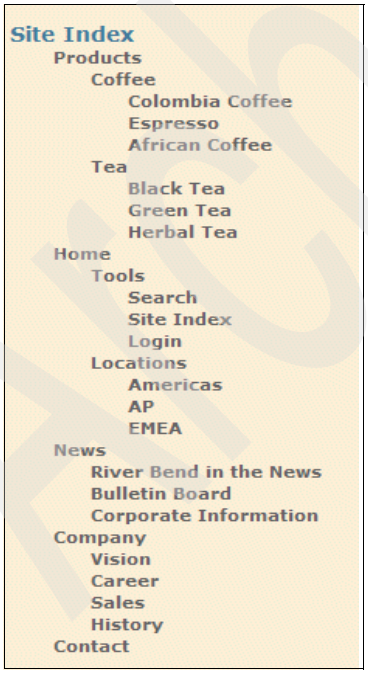


Figure 5-29 River Bend's site map

Archived

Menu component

A *menu* is ultimately a search mechanism that groups together related documents based on specific criteria such as categories, keywords, authoring templates, site areas, and so forth.

To users, menus and navigators provide similar functionality, which allows a user to navigate around the site or to locate specific content. A menu differs from a navigator in the way that t is constructed and in the way that it displays the resulting links. A menu displays content, while a navigator displays site areas for navigation. Think of a menu as a *list* of documents grouped by a number of different criteria (categories, site areas, keywords, and so forth).

When a document is created, changed, or deleted, menus are updated dynamically, and the link lists displays the recent changes, eliminating the need for re-coding and preventing the possibility of broken.

Important: Menu-landslides are updated automatically to reflect changes in the content library. Their consistency is maintained by the system.

Menus display links as images, icons with text, links with a summary, or many other combinations (see Figure 5-30 on page 183).

Archived

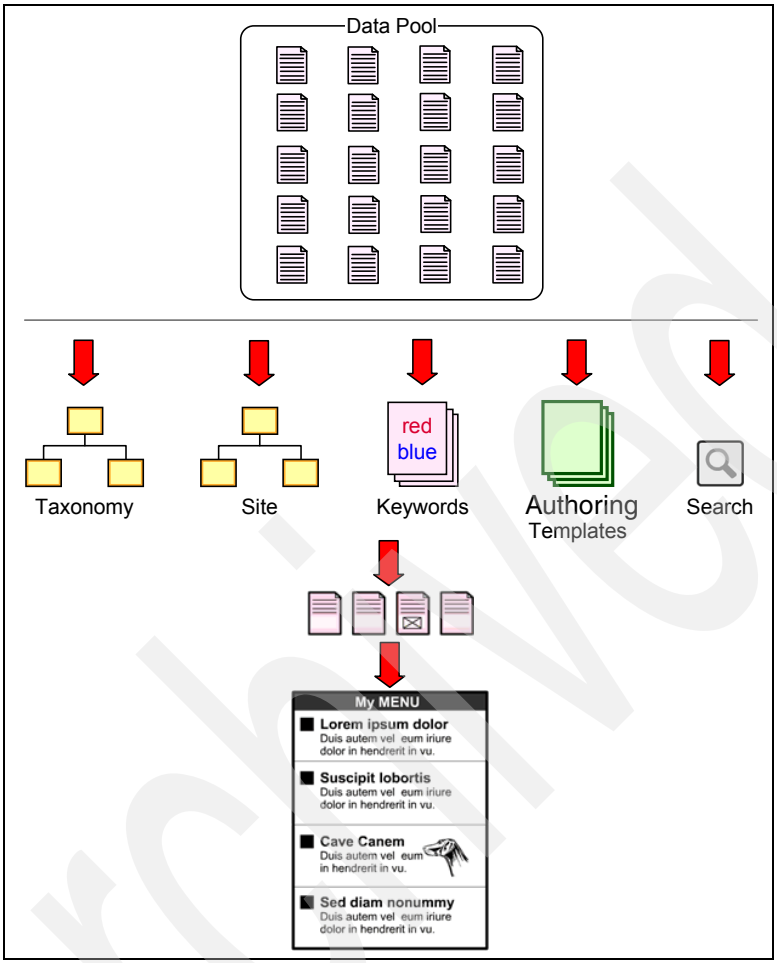


Figure 5-30 Menu construction from various selection criteria

Archived

The River Bend Web site displays content that is created for the River Bend in the News site area through the menu *MENU-Site Area* (Figure 5-31). The menu displays the display title, creation date, summary, and a link to open the document so that the user can read the entire content.

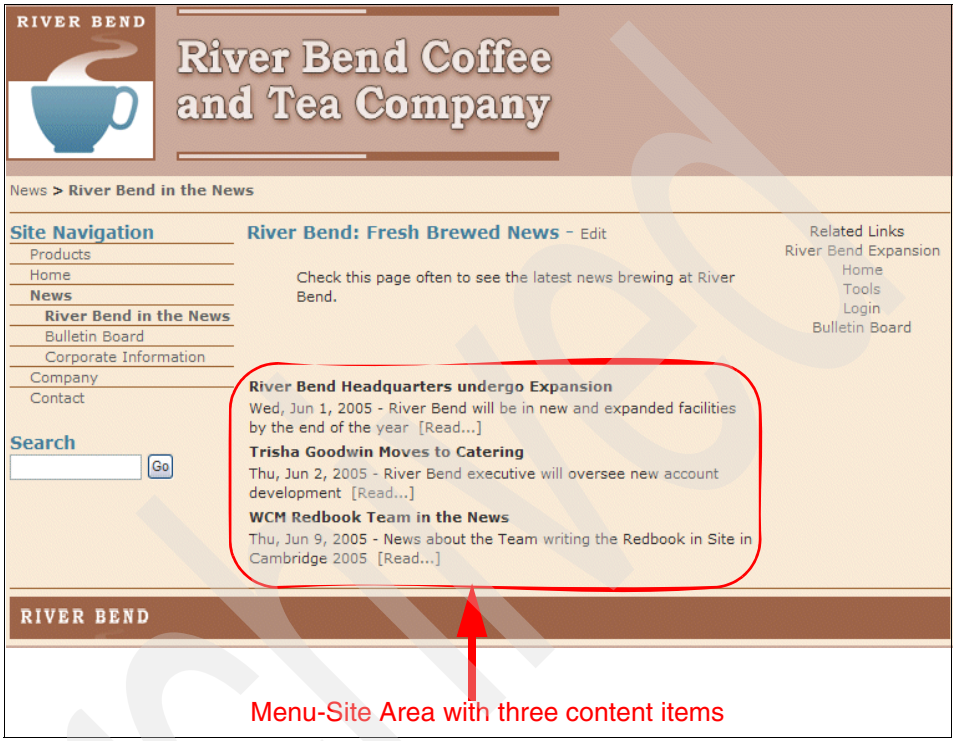


Figure 5-31 MENU-Site Area on River Bend in the News page

The following options define a menu using the menu MENU-Site Area for search criteria:

- The available selection criteria can create very granular menus. In the River Bend Web site, the *MENU-Site Area* definitions are kept very general.
- Matching content associated site areas define a menu for content that is linked to the selected site areas using the *Select Site Areas* button. This action makes the menu specific for a certain site area but not generally usable as the MENU-Site Area is intended. It has to be usable for every site area in the River Bend Web site.
- Similar to navigators, ancestors and descendants can be included in the menu selection by select the *Include all Ancestors* and *Include all Descendants* options.

Archived

- The Matching Authoring Templates option — used in the MENU-Related Links — identifies content that is created using the authoring template that is selected for this option. This option is not related to the hierarchical site framework structure but finds content beyond and independently.
- Matching content that is associated with categories takes advantage of content categorization. Again, independent from the hierarchical site framework, all content is found that contains the categories as specified in this field. Selected the results must match all categories option creates a AND link between all categories.
- Identical functionality as matching content that is associated with categories provides the *Matching Content Keywords* field. However, the keywords have to be typed rather than selecting them from a list. The related field in the authoring template — whose values are compared in this option — is also a text field where users enter the keywords. Finding content with this option can be difficult as no guidance on keyword spelling nor a list of existing keywords exists.
- Identical to these options, the *Matching Profile Search Rules* option allows the user to select search rules as identification criteria for content. (For information about search rules, see 5.3.10, “Search rules” on page 194.) By default available options are:
  - Current Content Categories:

    The menu displays content profiled with the same Categories as those used to profile the current content.
  - Current Content Keywords:

    The menu displays content profiled with the same Keywords used to profile the current content.
  - Current User Categories:

    The menu displays content profiled with the same Categories as those selected by the current user
- For the *MENU-Site Area*, all the values are empty.
- Selecting the *Only Include current Site Area* option finds all content items that are linked to the current site area which are activated by the user by clicking an element in the navigator NAV-Left.

The following options define the presentation of content using the menu MENU-Site Area:

- The current document can be included in the displayed menu by selecting the *include current document in results (if applicable)* option. The *if applicable* text reminds you that a menu can be displayed in a content context and also can be based on a site or site area.

- The ascending order for results option is selected. Alternatively, deselecting the field creates a descending order based on the sort key of Results Primary, Results Secondary, and Results Tertiary sort key fields. Available sort keys are: name, description, published date, and last modified date.
- MENU-Site Area items are sorted first by publish date and then by name and tertiary bay description. This selection shows the newest document at the end of the list.
- The results per page is set to zero (0) to allow all items to display on one page regardless of how many are found. Increasing the number changes the number of documents that display. For example, changing the results per page to 3 displays documents 1 through 3 on page 1, documents 4 through 6 on page 2, and so on.

  For a better overview, limit the number of displayed documents to no more than 10. Also, sort in reverse order so that the newest document is at the top of the list.
- The Start Page field defines on which page results display.

The following fields define the design for the menu to appear on the pages through text, tags, and components:

- Header Design can specify a header text or component to appear before the menu list. The MENU-Site Area header is empty.
- The appearance for the Menu elements are defined in the Component Design for each matching Content field. The MENU-Site Area uses a HTML Component reference:

  ```
  <AptrixLibCmpnt name="HTML-Layout-Standard Menu"/>
  ```
- The footer defines the design below the menu and separator between the menu element. However, for the MENU-Site Area, these fields are empty.

Archived

Figure 5-32 illustrates these options.

MENU-Site Area

* Required Fields.

Identification

Menu Component

Matching Content associated Site Areas:
-- Select Site Areas

Include all Ancestors:

Include all Descendants:

Matching Authoring Templates:
-- Select Authoring Templates

Matching Content associated Categories:
-- Select Category

Results must match all Categories:

Matching Content Keywords:

Matching Profile Search Rules:
-- Select Profile Search Rules

Include current document in results (if applicable):

Include current Site Area:

Ascending order for results:

Results Primary Sort Key:
Publish Date

Results Secondary Sort Key:
Name

Results Tertiary Sort Key:
Description

* Results per Page:
0

* Start Page:
1

Debug Mode (Java Console Only):

Figure 5-32 Definitions and values for MENU-Site Area

A second menu, MENU-Category, which we used in Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253, uses the Espresso category as search criteria. This category allows the user to find content throughout the Web site independently of a site area. (The menu concept allows the combination of categories and site areas as search criteria.)

Archived

The MENU-Category is referenced in the Espresso site area rather than in a content item. In this case, the *MENU-Category* appears on the Espresso site area page and displays content that has the category *Espresso* (title of *Rich Colombia Coffee from West coast Plantation*). At the same time, the same MENU-Category is linked to the *Colombia Coffee* site area and displays on the Colombia Coffee page as well through the Menu-Site Area. Thus, the same content displays by two different menus that use different search criteria.

Text component

You can use text components to store text or HTML as reusable text modules. For the River Bend Web site, you can create text components to describe standardized processes for coffee and tea production, such as roasting or picking. (Tea picking is different from coffee picking.) You can use the text components as standard text blocks in the Coffee and Tea product description pages.

Rich Text component

The Rich Text component is very similar to the text component. However, Rich Text components can reference one or more components using tags and also provide a Rich Text editor for formatting text and HTML. You can use a Rich Text component on the River Bend Web site on the Coffee and Tea product pages to provide a standardized purchase order form on each page. The Rich Text component reference the form and its components (design, buttons, and so forth).

File resource component

File resources are components that provide a central repository for files, images, stylesheets, and so forth. Authors and site designers can reference the file resources. The River Bend Three Column Table presentation template references the Riverbend.css stylesheet as a file resource component as shown in the following example:

```
<link rel=”stylesheet" href='<AptrixLibCmpnt name="Riverbend.css"/>'>
```

Another example of a file resource component is graphics that are used to polish the River Bend Web site. The navigators do not use any symbols (such as arrows) to indicate activated sections. You could store a graphic as a file resource to serve as indicator for an activated selection.

HTML component

You can use an HTML component for almost any purpose. An HTML component contains HTML code and often references items, content fields, and components. The River Bend Web site uses the following HTML components:

- The navigator on the left side, NAV-Left, displays its elements in alternating designs. Activated elements appear in bold layout, while other elements use the default layout.

  The bold layout uses the *HTML-Layout-Main Nav Bold* component:

  ```
  <a href='<Placeholder tag="href"/>' >
  <b><Placeholder tag="name"/></b>
  </a><br>
  ```

  The default layout uses the *HTML-Layout-Main Nav Plain* component:

  ```
  <Placeholder tag="namelink"/><br>
  ```

  Figure 5-33 shows the result of applying these layouts.

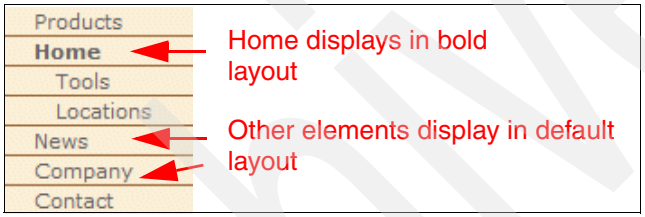


Figure 5-33 Navigator element defined in different HTML components

- The layout for the content, Menu MENU-Site Area, the *HTML-Layout Standard Menu* component to format the date, summary, and a link below the displayed title, as shown in the following example:

  ```
  <table cellpadding="0">
  <tr>
  <td>
  <b><AptrixCmpnt context="autoFill" key="Display Title" type="content"/>
  </td>
  </tr>
  <td>
  <WorkflowCmpnt context="autoFill" field="publishdate" format="EEE, MMM
  d, yyyy" type="content"/> - <AptrixCmpnt context="autoFill"
  key="Summary" type="content"/>
  <a  href='<placeholder tag="href"/>'> [Read...]</a>
  </td>
  </tr>
  </table>
  ```

Archived

Figure 5-34 shows the result of this component.

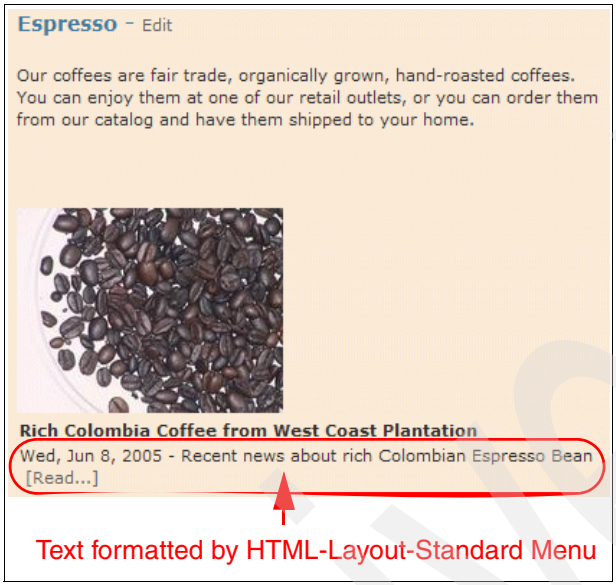


Figure 5-34 Text in Menu formatted with HTML component

Image component

You can insert images that are used throughout the Web site directly into image components fields in the authoring templates. In this case, the author has the option to define the size of the image, which might or might not fit the overall design. Defining images as *image resources* provides identical images for reusage.

Archived

The River Bend Web site uses five image components as consistent design elements in the presentation templates. These are the logo, the River Bend title, the two brown lines, and a footer, as shown in Figure 5-35.

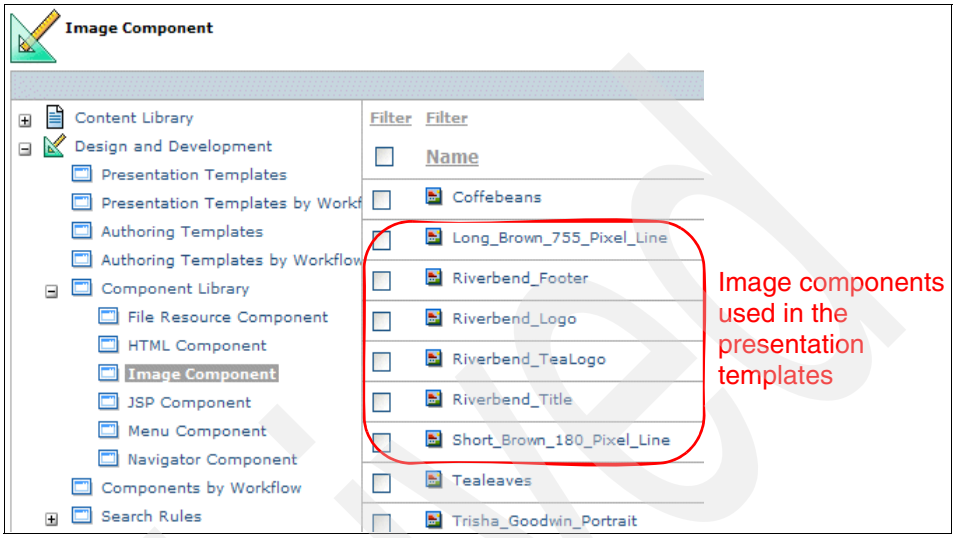


Figure 5-35 Image components in component library

The other images are used in content in the PostBody field. You can reuse the images easily as with *Coffeebeans* for all Coffee-Site Area and *Tealeaves* on all Tea-Site Area related children's pages, as shown in Figure 5-36.

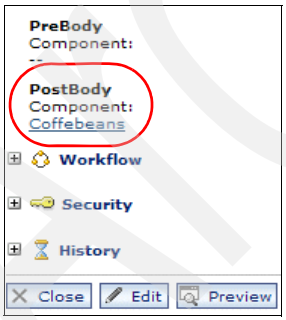


Figure 5-36 Image component used in PostBody field

Archived

JSP component

JSP components store the path to a JSP and the content of a field for an error message in case the JSP is not available. A extended discussion on JSP in combination with the API is found in Chapter 12, “Working with the application programming interface” on page 469 and Chapter 13, “Advanced topics and examples with the API” on page 533.

Document Manager component

The *Document Manager* component is only available for WebSphere Portal 5.1 and requires integration to Portal Document Manage. The Document Manager references a document (that is, a file) that is stored in a Portal Document Manager library.

> **Restriction:** This option requires IBM Workplace Web Content Management to run on the same server as Portal Document Manager.

To create the component, the author has to have access to Portal Document Manager, the library, and the document. Select the document in the interface as shown in Figure 5-37.

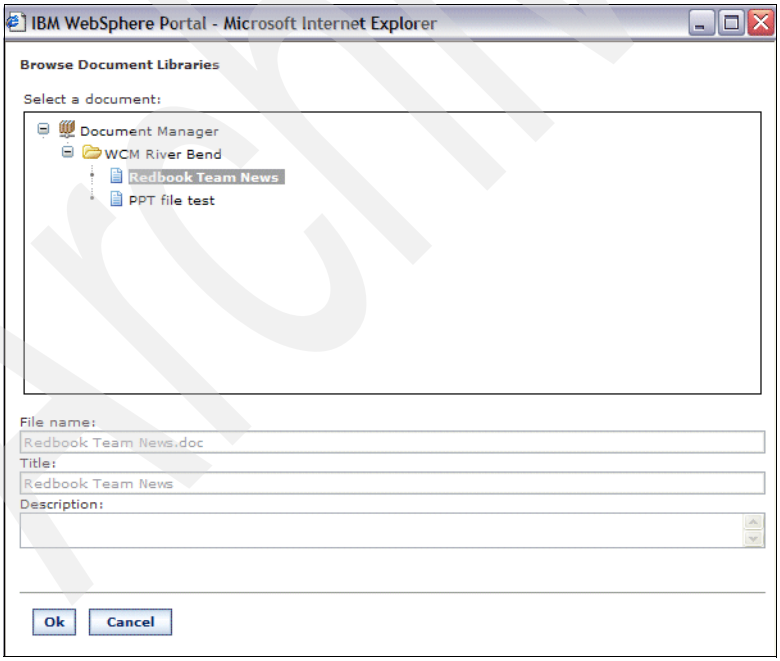


Figure 5-37 Selecting a document in Portal Document Manager

IBM Workplace Web Content Management references the path to the document in the component, as shown in Figure 5-38.

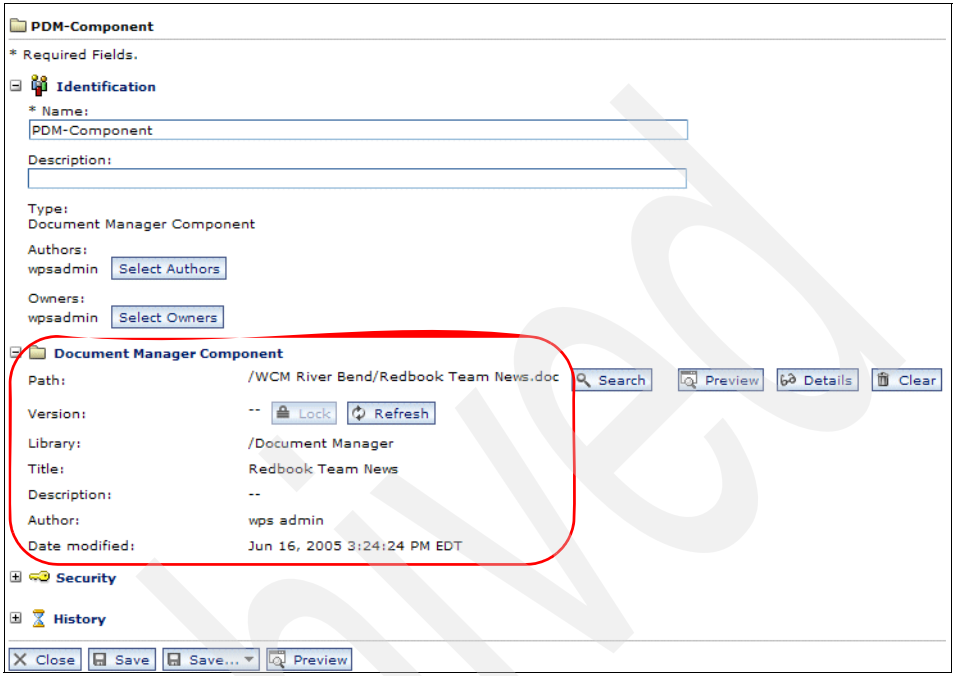


Figure 5-38 Portal Document Manager path reference

The Portal Document Manager component can be referenced in the same way that other components are referenced. You can use it to show not only content but also a URL to the Portal Document Manager file or an attribute of the file.

To reference the component, use the following code:

```
<DocumentManagerLibCmpnt convertTo="text/html" name="<component name here>"
scope="Document"/>
```

For other file formats, you need to adjust the `convertTo` attribute (for example to BMP or JPEG format).

Taxonomy component

Taxonomy components relate often to personalization where lists of categories are presented to the user to select categories of interest. Alternatively, the you can use the taxonomy component to display categories for navigation instead of site areas. However, be aware that this option works only for authenticated users.

Archived

If your site requires this option for anonymous users as well, the implementation is more complex. In this case, you need to create a JSP for an imaginary user for which the component is displayed.

The taxonomy component displays a list of categories, starting with the selected start area (in this case, a category), and displaying the depth according to the Depth field. You can define further selection criteria by using a Profile Search Rule.

You can use the remaining fields to format the displayed categories with a header and footer. The Unselected Components and Selected Component fields contain special layout for selecting categories, which could be checkboxes.

User name component

You use user name components to display components for anonymous users or authenticated users and to create different design experiences based on user names. The logic for a user name component is as follows:

If an anonymous user is using the site, then it displays the Anonymous User Design, or else the User Component Design, where *Anonymous User Design* and *User Component Design* are the fields that are available in the component form. These fields contain designs that are based on HTML code or component references.

A good example for a user name component is a welcome message on a Web site's home page for an authenticated user. The anonymous user does not see a welcome message.

5.3.10 Search rules

Search rules are used in menus as search rules for identifying content. By default, the following search rules are available:

- Current content categories:

  Finds content profiled with the same Categories as those used to profile the current content.

- Current content keywords:

  Finds content profiled with the same Keywords used to profile the current content.

- Current user categories:

  Finds content profiled with the same Categories as those selected by the current user

You can further restrict search rules by creating new search rules beyond the restrictions of the default rules. Available types of search rules are *Restricted Current Content Categories*, *Restricted Current User Categories*, and *Query String Categories*.

Restricted Current Content Categories Search Rules and Restricted Current User Categories Search Rules allow you to restrict search to a certain branch in the taxonomy tree beyond the categories that are defined in the current content or that are defined by the user who uses a taxonomy component. (See “Taxonomy component” on page 193 for more information)

The interface provides only one field to select the restriction, as shown in Figure 5-39.

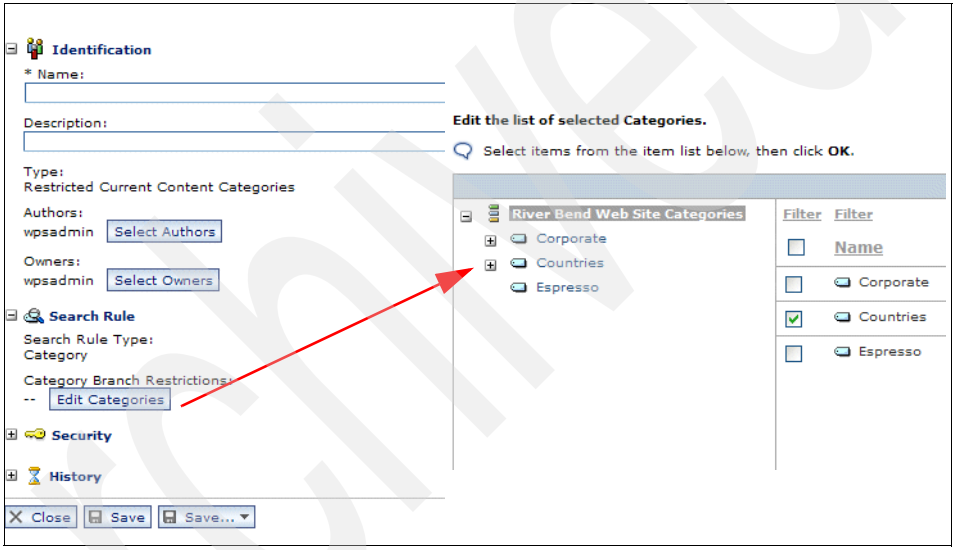


Figure 5-39 Select restriction categories for search rule

Archived

5.4 Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1

Portal Document Manager allows teams, groups, and individuals to set up libraries and folders to manage documents to a certain extent. Although Portal Document Manager does provide some document management capabilities, it is *not* a document management system or enterprise document management system such as Content Manager, DB2 Document Manager, or Lotus Domino Document Manager.

Attention: You can use Portal Document Manager files based on the implementation of the JSR 170 specification that is available with WebSphere Portal 5.1. However, functional limitations apply for IBM Workplace Web Content Management 5.1.

Portal Document Manager documents can only be referenced from the same WebSphere Portal server on which IBM Workplace Web Content Management is installed. Thus, Portal Document Manager files that are referenced from a Portal Document Manager component within IBM Workplace Web Content Management are *not* included in Syndication! If Portal Document Manager files must be available in a distributed environment, you have to develop a process to move Portal Document Manager files from one server to another. You can use the Portal Document Manager export and import functionality to achieve this development.

Because JSR170/JCR was not completed until the end of May 2005, further portal versions will continue the implementation of JCR and will enhance its functionality.

Portal Document Manager integration is available only for IBM Workplace Web Content Management 5.1. There is no installation necessary. However, to use a

Portal Document Manager document within IBM Workplace Web Content Management, you need to do the following:

- In Portal Document Manager, you need to:
  - Create a Library.
  - Create a Folder.
  - Create Documents.
- In IBM Workplace Web Content Management, you need to:
  - Create an authoring template with a Portal Document Manager component. The fields in the new authoring template will match the existing presentation template component references.
  - Create content with the new Portal Document Manager authoring template.

Attention: The Workflow features in Portal Document Manager and IBM Workplace Web Content Management 5.1 are completely independent and do not interact. You should use Portal Document Manager Workflow to approve content that is stored in Portal Document Manager, and you should use IBM Workplace Web Content Management Workflow to approve publication of documents to the Web site.

For detailed information about how to integrate Portal Document Manager with IBM Workplace Web Content Management 5.1, see 7.26, “Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1” on page 341.

Archived

Archived

6

Information architecture and site design

When designing a Web site, you should organize information in a way that is readily accessible to users. The structure in which information is organized is called the *information architecture*. This chapter provides an approach for determining the information architecture for the top levels of a Web site. You structure the information that is published on a Web site within a *site framework* that is accessed through Web site navigation. This chapter leads you through the decision-making process by discussing and defining the information architecture, key considerations and decision processes, and information design. Additionally, it discusses how to define criteria for site acceptance and to provide support for a workshop approach.

It is important to acknowledge the relationship between the information architecture and the site framework. The site framework structures the information about a published site and is closely integrated with the Web site navigation. (See 5.3.8, “Site framework” on page 170 for additional details on the importance of the site framework.) To begin, you define an initial site framework with primary and secondary site areas. The site framework is a prerequisite for the design and development of the content management system. In addition, you need to develop an initial layout of the home page based on the primary and secondary site areas. Both the framework and the home page layout require review and validation — with business stakeholders and users — from both an authoring and Web usability viewpoint.

© Copyright IBM Corp. 2006. All rights reserved.

6.1 Defining the information architecture

Content accessibility on an intranet can have a direct impact on an organization's overall productivity. If users can find the information they are looking for before they even are aware that they are searching, then you have accomplished the goal of organizing the information correctly.

The primary reason to share information about a Web site is to make it available for users. This fact sounds incredibly obvious. However, it is often the central reason why Web sites fail for the following reasons:

- Information is often voluminous and widespread over different divisions within an organization, making it difficult to maintain a consistent oversight of the structure of information and to determine a structure that is easy to understand and to use.
- There is rarely a single person or department with a consistent oversight of all the information that an organization wants to share.
- Information is something used everyday, and as a result, it is often not obvious which information is most valuable or how it should be structured.
- Using a content management system encourages content providers to share more information while getting more experience, resulting in a strain on the primary organization of the content.

The information architecture defines how the information about a Web site should be organized and linked so that users can access content. An organization should investigate, analyze, design, and implement the information architecture for a site. Then, the challenge is presenting a image that enhances how the user experiences the information.

You might find reasons why a specific path to information supports your business needs. The experience a user gets can be the key to success and often reflects the organization’s philosophy. Frequently, an organization builds its success on a unique customer experience that cannot be ignored when architecting a Web site. In addition, the audience can vary such that it becomes necessary to separate users into categories to get a good acceptance of the way information is presented on the Web site. For example, consider the following diverse offerings:

- A food company might offer both traditional, home style food as well as more contemporary fast food.
- A vehicle manufacturing company might have diverse offerings which range from passenger cars, to industrial trucks, to motorcycles (BMW).
- A technology company might offer a range of products and services, from technical consulting services to consumer electronics (Hewlett Packard).

Because of this diverse range of products, brands often get their own Web sites with there own information architecture and Web address (URL). In this case, a master (parent) Web site which includes links to detached Web sites representing the specific branding are the best choice.

Defining the information architecture is often the most underestimated part of a content management project. Organizations frequently spend a lot of time finding the right content management system and then deciding on the best information architecture for that system.

When planning the information architecture, an organization must determine:

- The hierarchical structure of the site
- The functionality that is required on the site
- The look of individual pages
- How to classify the content
- The flexibility of the architecture to allow the business to evolve

The information architecture determines the structure of the site, how navigation is derived, and the ease of navigating the site. You develop the following information architectures when designing a content management system:

- Site framework
- Category hierarchy
- Document type hierarchy

Defining the information architecture lays much of the groundwork for how content is organized on a site. Regardless of where the content resides, you need a good understanding of the content that is to display.

Tip: A good information architecture team includes representatives of all content areas that are included on the Web site. It is also beneficial to have a single person lead the team who is willing to take a leadership role and to make the required decisions.

6.1.1 Base example of a good information architecture

The section presents some examples of a good information architecture and discusses what makes these examples special.

When considering the information to share on a Web site, it is good to have direct access to the top levels of content. These top levels are the logical entrance to information and build the foundation of the information architecture. There are no strict rules governing what these top levels should be.

Archived

The River Bend Coffee and Tea Company Web site

The River Bend Coffee and Tea Company is a virtual company that we use for this redbook to give a simple example of a basic information architecture. This fictitious company sells coffee and tea in different locations around the world. They have a public Internet site to share information with their customers. These customers can be individuals as well as businesses.

River Bend identified the following top levels of site areas or containers for content that are directly accessible from the home page:

- Products
- Company
- Locations
- News
- Contact

From these top levels, a secondary site area allows users to specify what topic they are looking for within that area. For example, the secondary site area for *products* includes *coffee* or *tea*. This architecture provides an obvious separation that users can comprehend and that leads users to exactly what they expect to find.

The third level site area offers more content on the chosen topic. For example, under *tea* users can find *black tea*, *green tea*, and *herbal tea*.

Figure 6-1 illustrates the complete hierarchical structure of the River Bend Coffee and Tea Company Web site.

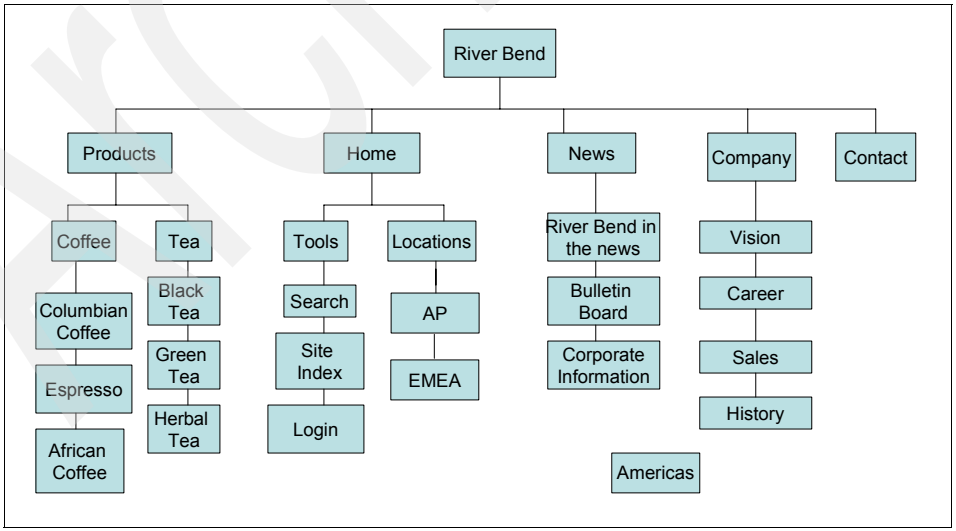


Figure 6-1 River Bend Coffee and Tea Company navigation

The IBM developerWorks Web site

IBM developerWorks is a Web site for technical-oriented users of IBM products. It is a resource for developers. IBM developerWorks offers highly technical information and additional software that is related to all IBM products. The site encompasses a broad scope of content. Visitors to the IBM developerWorks Web site typically have a specific need to find help or the latest news for a particular product. Most time, theses visitors also expect quick answers. Providing this level of support on the IBM developerWorks Web site is a challenge for the information architecture, because the site contains multiple thousands of content pages.

To provide an example, let us look at the structure for the information architecture of the IBM developerWorks Web site. IBM developerWorks is segmented into the following areas:

- Top tool bar

  The top tool bar for the IBM developerWorks Web site contains a regional selection, terms of use, a search function that is separated by area, and the main IBM navigation menus (Figure 6-2). This top tool bar is available on all IBM Web pages.

Figure 6-2 Top tool bar

Archived

- Top-level navigation of IBM developerWorks

  From the top-level navigation, users can reach all product content. The navigator is split into the following parts: products, topics, and feedback (Figure 6-3).

Figure 6-3 Top-level navigation of IBM developerWorks

Depending on the browser technology, visitors have either a dynamic opening second level under each topic or menu options that they can reach by clicking. The secondary site areas are similar for all products. Each contains all related information for that product. Additionally, secondary site areas are structured by relevance to users, ranging from a new user to an expert user. So, the IBM developerWorks Web site uses additional features of the information architecture — *relevance* and *consistency*.

The third level navigation opens above the main content area in the center of the page.

Archived

- Menus

  Depending on the selected navigation level, the right area of each page shows menus as shown in Figure 6-4. (These menus are discussed in detail in 6.3.3, “Using menus” on page 238.) These menus give users one-click access to related content or offer additional features.

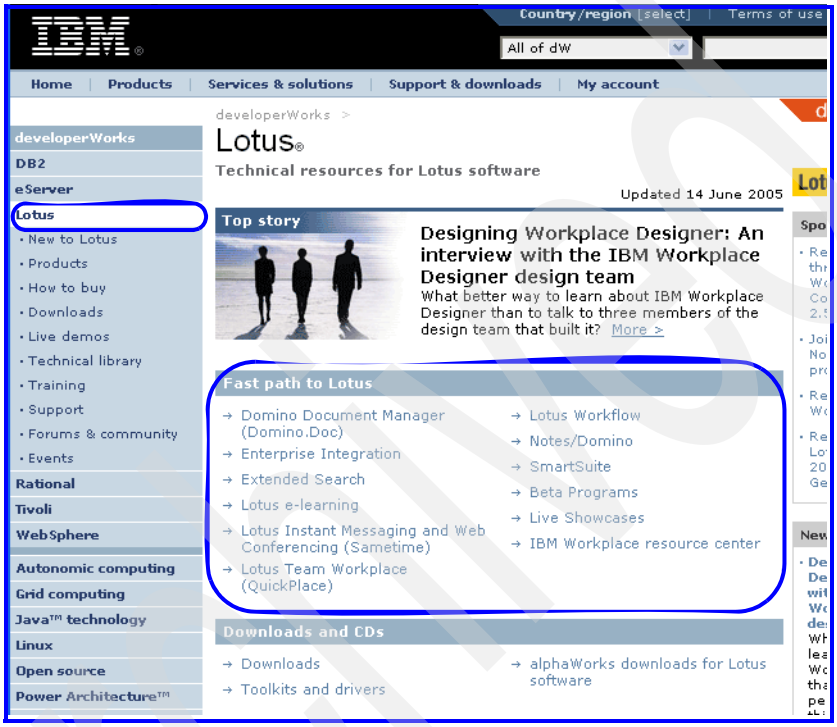


Figure 6-4 Menus

- Main content area

  The center of each page displays the content. Links to further information enrich the content displayed in the main content area.

This structure makes IBM developerWorks a very good example of an information architecture. This information architecture covers several needs, including intuitive navigation, consistent structure, and a user friendly site for both the first time visitor and the frequent user, in a way that serves the wide scope of various content.

The AO Foundation Web site

Our last example is the Web site that won the IBM Lotus Award 2005 in the category of Best Portal Solution. That Web site is owned by the AO Foundation in Switzerland (http://www.aofoundation.org). We include this Web site as an example because it shows an outstanding information architecture in the area of musculoskeletal injuries. Even though this topic is highly specialized, the information architecture of the site helps the visitor get the information that is required. This example of a highly complex topic that is presented in an intuitive way can be transferred to many different business types.

The AO Foundation is a non-profit organization that is dedicated to improving the care of patients with musculoskeletal injuries and their sequela.[1] The goal of AO is the research, development, education, and quality assurance in the principles, practice, and result of fracture treatment. They structure themselves internally into several institutes and projects that are governed by several boards. The goal of the AO Foundation is the research, development, and approval of every aspect of osteosynthesis methods and devices and to provide the education to spread this knowledge around the globe. Every day somewhere in the world an AO Foundation education class starts. Using these methods and devices, the surgeons and personnel trained in the AO have successfully treated a vast quantity of patients with musculoskeletal injuries and their sequela over the past 40 years. However, information overload was making it increasingly difficult for surgeons to identify and access relevant knowledge quickly. To improve member service, AO Foundation wanted to make the right information easily available to surgeons.

> *Information overload is a huge problem for surgeons. Publications of all kinds help them keep up with developments, but the relevant knowledge to be applied in a clinical situation is often buried and hard to find when needed.*
>
> Michael Redies, Head of Knowledge Services, AO Foundation

Now that we understand the content and the need of the AO Foundation, let us take a closer look at how to fit this information within a successful information architecture. AO decided to develop a comprehensive Web portal that gives members easy, secure access to the organization's breadth of knowledge. Rather than simply transfer written content verbatim to the online environment, AO revamped the information, making it more applicable from a clinical perspective. For example, AO grouped together all of the films, lectures, studies, and chapters relating to various surgical procedures into their own separate, structured repositories. This way, they can quickly lead surgeons through the different steps of an operation rather than making them find and then match sense of disparate pieces of information.

[1] *Sequela* is a pathological condition resulting from a prior disease, injury, or attack (for example, a sequela of polio).

The portal also provides a powerful search engine so users can find the precise information they need from AO's extensive libraries. After initial sign-on, members can also conveniently browse through communal areas while built-in security measures safeguard the research forums from unauthorized access.

On the entrance page, the surgeon finds an overview of the currently available content, structured by a human skeleton (Figure 6-5). This choice option is the top-level navigation and is so intuitive to use that even non-medical, first-time visitors are able to take the first step. This design is a good example that a navigator does not need to be in text form.

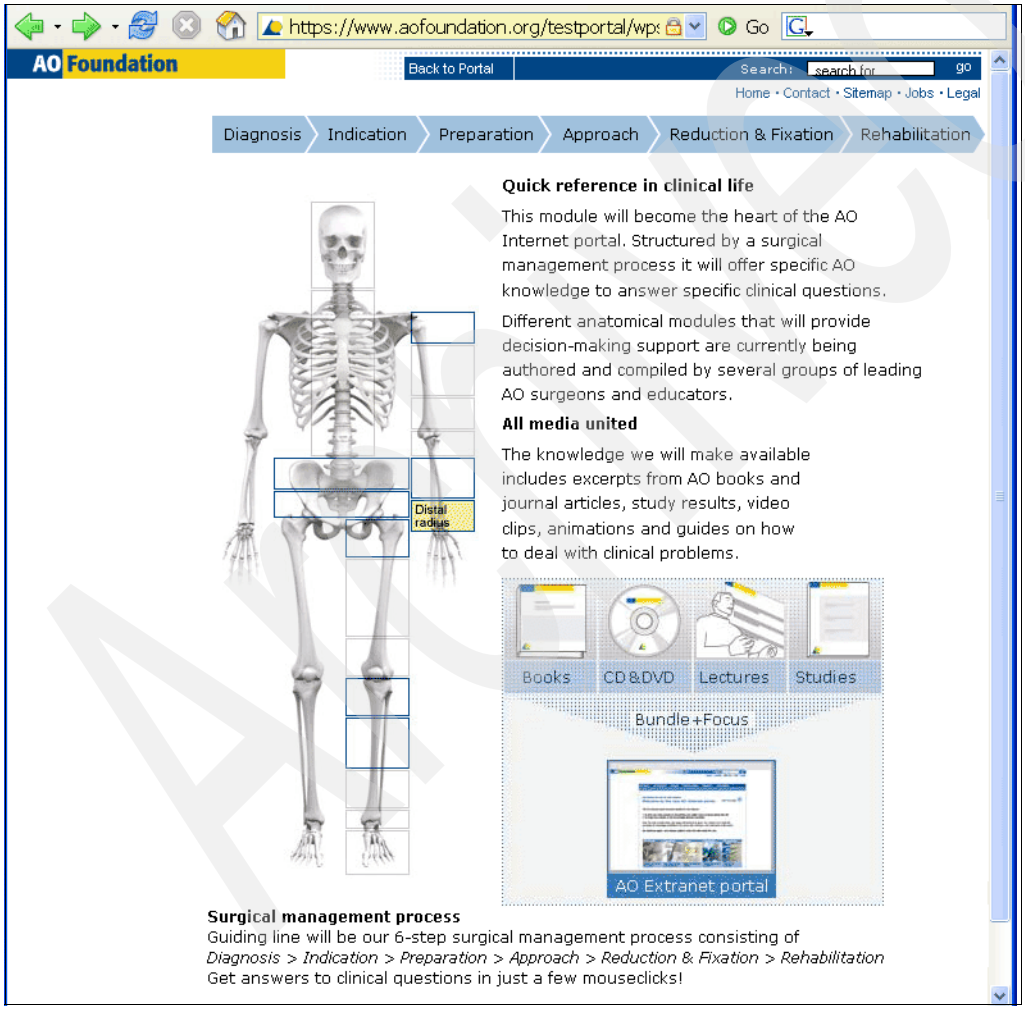


Figure 6-5 The entrance page of the AO Foundation Web site

The second level area is *Diagnosis*, which leads into the diagnosis of fractures. Because bones can break in various ways, the site includes a classification of fractures. When users select a certain bone segment (in our example, we select the wrist — the Distal radius), the page is then segmented into four areas, as shown in Figure 6-6.

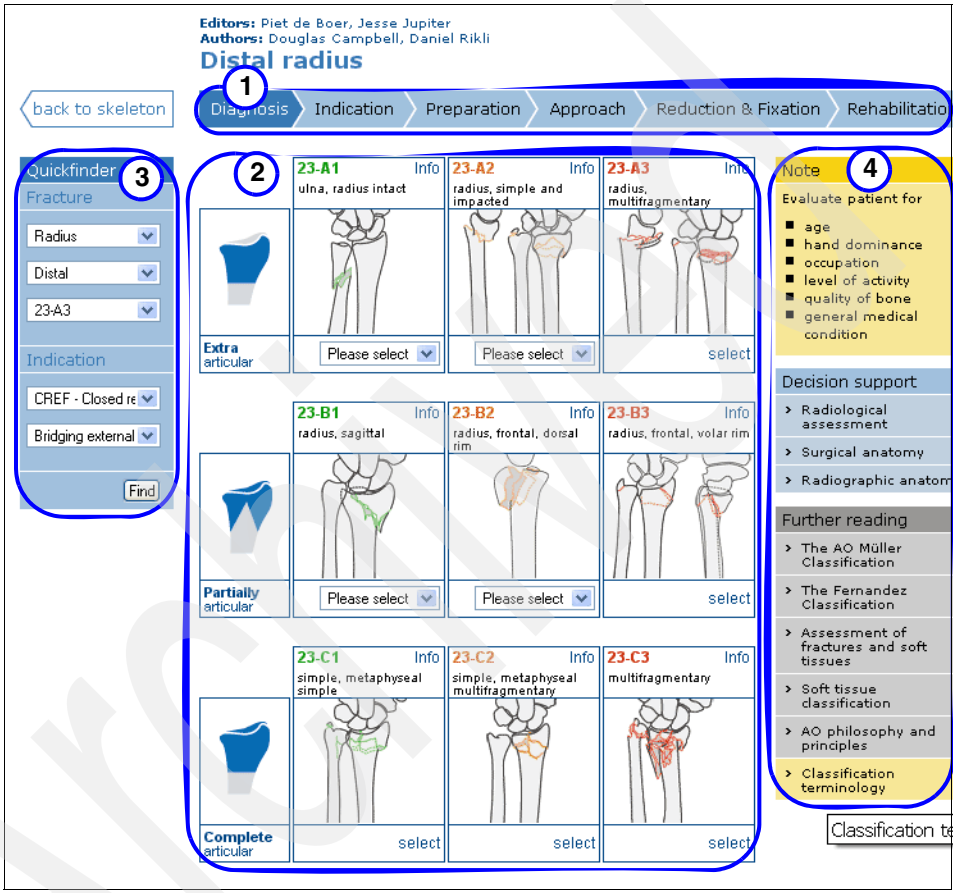


Figure 6-6 AO Foundation second-level navigation

The four areas include:

- Top-level navigation

  The navigation bar on top includes the steps that a surgeon follows when planning a bone fracture therapy.

- Classification navigation

  The classification navigator is not obviously a navigator. It leads to additional information about a specific topic, as shown in Figure 6-7.

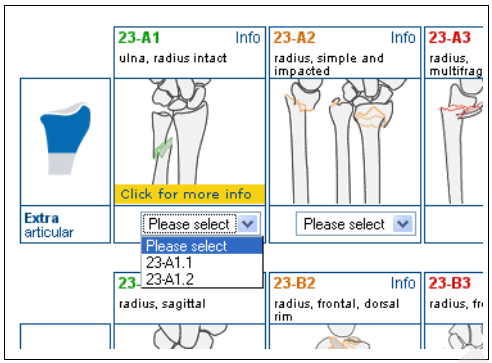


Figure 6-7 AO Foundation classification navigation — additional information

Clicking **Info** or a picture opens a window with additional information, as shown in Figure 6-8.

Figure 6-8 AO Foundation classification information window

Archived

Clicking **Select** leads to the next content level — the *Indication*. If the indication is selected before specifying the diagnosis (that is, the type of break), an information reminds the visitor to select the diagnosis first, as shown in Figure 6-9. This methods displays the full navigation path, but the visitor can only proceed in a given decision flow, using a controlled step-by-step lead through.

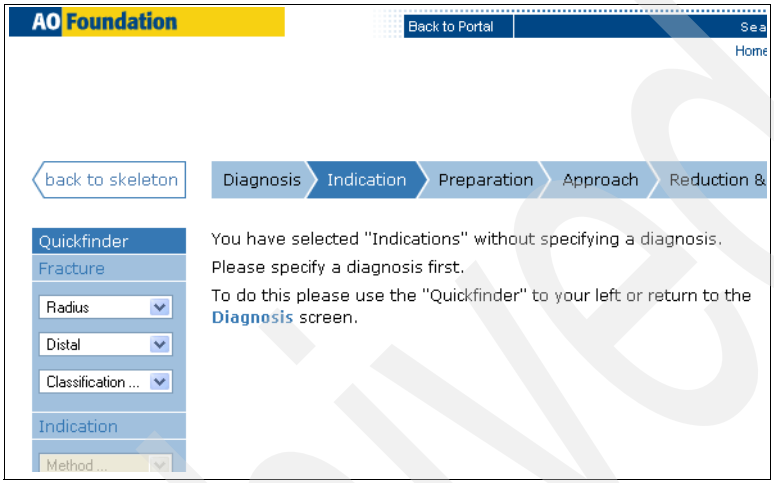


Figure 6-9 AO Foundation Diagnosis selection reminder

- Quickfinder

  The *quickfinder* enables advanced visitors to jump directly into *preparation*, *approach*, *reduction and fixation*, and *rehabilitation* instead of using the classification navigation (Figure 6-10 on page 211). This method is an excellent example of how to give advanced visitors direct access to information. The quickfinder is not a search tool. It supports and follows the controlled step-by-step lead through, but it skips assisting information.

Archived

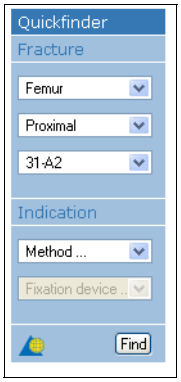


Figure 6-10 AO Foundation Quickfinder

- Note, decision support, and further reading

  This section offers additional information to the displayed content and menus, as shown in Figure 6-11.

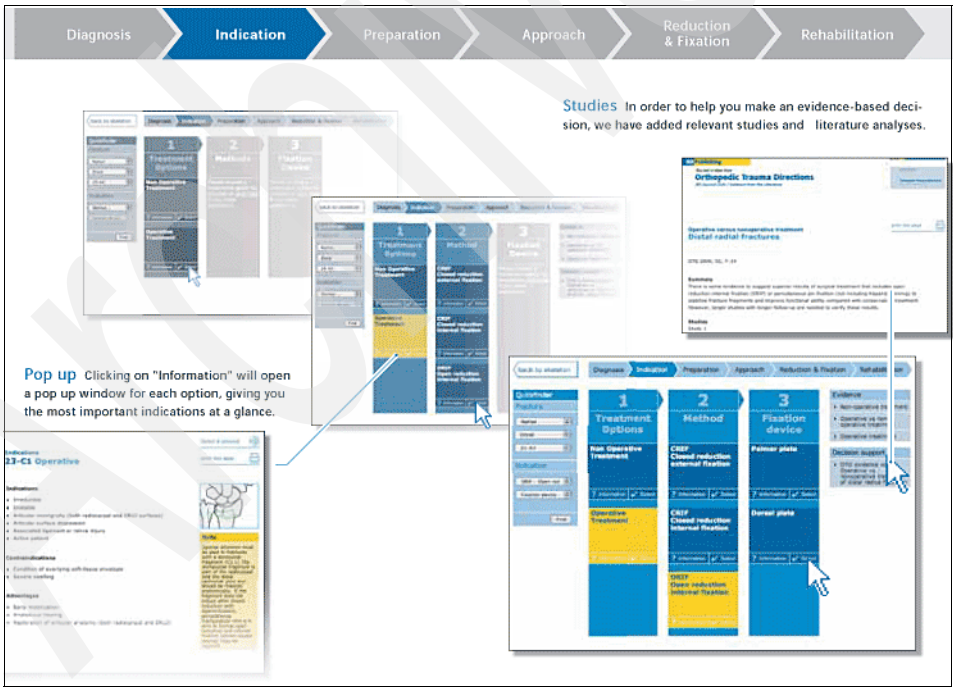


Figure 6-11 AO Foundation indication overview

Archived

The third-level navigation — *Indication* — is where the surgeon decides for a special treatment option, its method, and if necessary, a fixation device, based on the information selected within the Quickfinder, that is 1) Treatment Options (Figure 6-12), 2) Methods (Figure 6-13 on page 213), and 3) Fixation Device (Figure 6-14 on page 213).

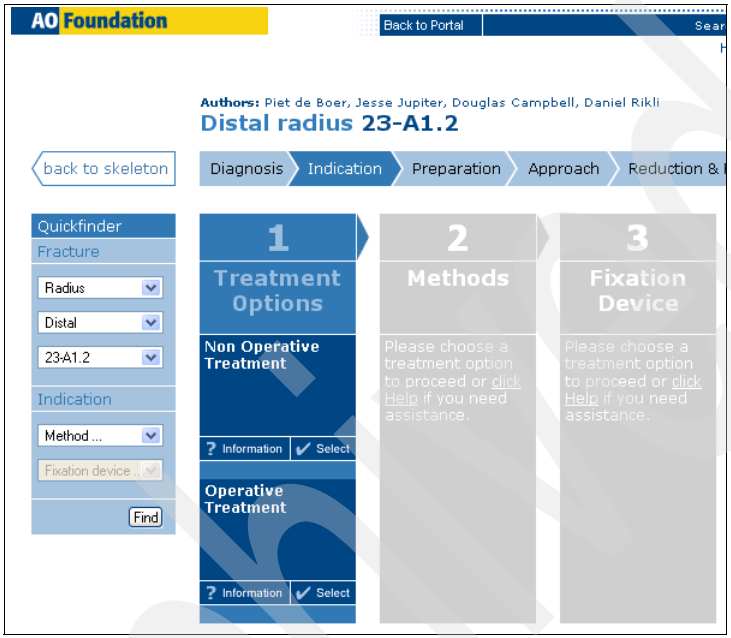


Figure 6-12 AO Foundation indication treatment options

Archived

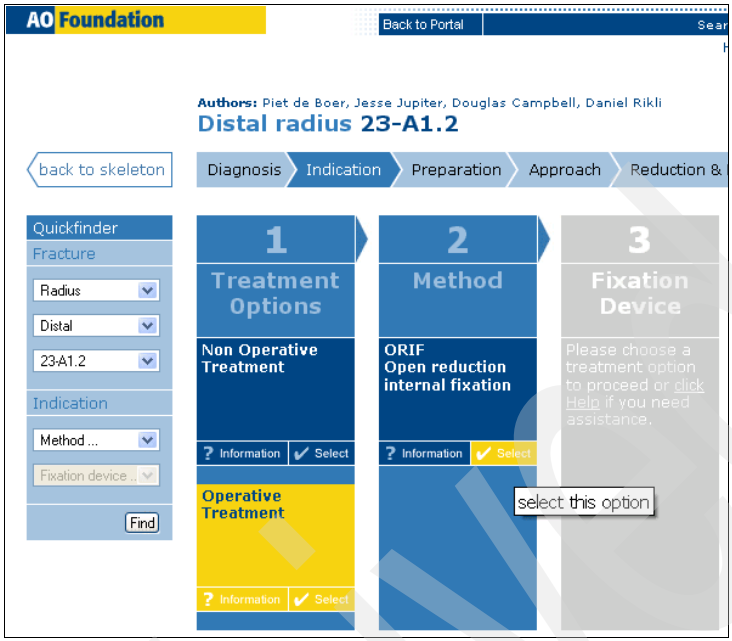


Figure 6-13 AO Foundation indication method

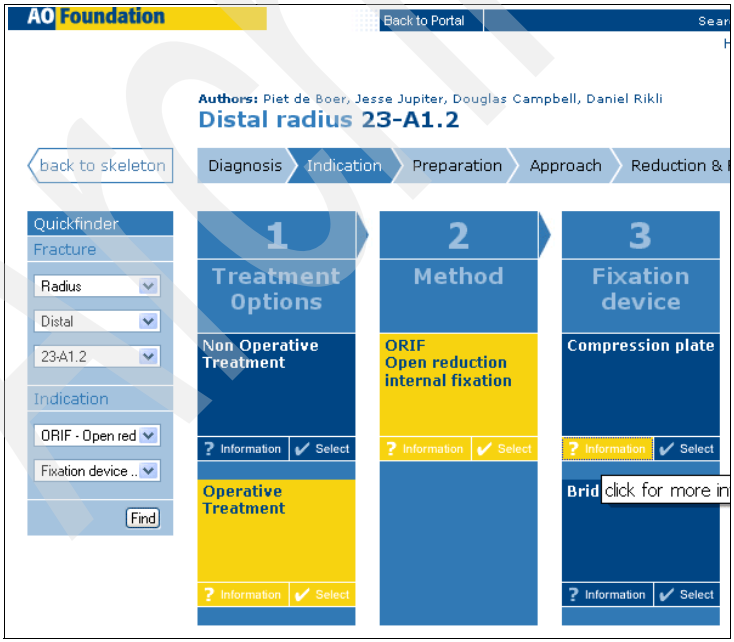


Figure 6-14 AO Foundation indication fixation device

This information architecture enables AO Foundation to include a further decision path within this level without leaving or complicating the overall navigation. Thus, the navigation stays intuitive and follows the decision path. Similar to the *Diagnosis* level, the *Indication* level offers additional information about a navigation topic by clicking that information. For the visitor, it is extremely helpful to know more if required before moving on and perhaps being mislead.

In these figures, we chose an operative treatment (ORIF, Open reduction internal fixation) and a compression plate. The indication navigation stays until we select the last step, Fixation device, Then, the navigation moves to the *preparation* level, as shown in Figure 6-15.

Figure 6-15 AO Foundation preparation

Preparation, *Approach*, *Reduction & Fixation*, and *Rehabilitation* reflect the fourth level of navigation and show content that is based on the navigation that the visitor selected previously.

The AO Foundation Surgery Module is an exceptional example of the information architecture of a highly complex navigation that supports a decision path covering thousands of content documents in an extraordinary and intuitive way. The information architecture also leaves options to expand the content on the Web site.

6.1.2 Results of an inadequate information architecture

Now that we have discussed some examples of good information architecture, this section discusses the consequences of using an inadequate information architecture.

Any Web content management project should point out the results of poor information architecture. For example, an inadequate information architecture shows up as inconsistent Web site structures and a poor user experience. Web sites that spread information over several areas do not follow a consequent architecture. In these cases, users might quickly lose the overview information and then they cannot reproduce the path from which they came. Another common issue with an inadequate architecture is the misapprehension of the user's experience. First time users have a totally different perception than the everyday content owners, where power users expect to access their information needs directly.

At times, an organization does not realize that the design of the information architecture is inadequate until the project is in progress. This can be the direct result of an ongoing analysis of an existing information architecture, or the opportunity to rethink how information is structured. In such cases, projects identify significantly more information and content than originally expected. Why and how does this happen? In many cases, this is the result of content owners rethinking which information should be surfaced. Additionally, content owners are now able to analyze information and content outside of the context of a pre-existing information architecture or navigation structure. Ultimately, this can foster new creativity and might expose the need for a new or more expandable information architecture. If the information architecture is not expandable, an inadequate information architecture is often the result. Either of these reasons can easily cause a Web content management project to be delayed significantly.

Internet or intranet Web sites are stand-alone content sources that use a structured information architecture. In this type of environment, the information architecture forms the foundation for how users consume the content — users visit the Web site to read and find information. Interaction with the user is limited to forms that the user submits. In the case of Portals, the information architecture in this environment needs to support content as well as applications in an integrated manner. In this case, an inadequate information architecture results in a break of information flow.

It is also possible that an organization can use an inadequate information architecture to force users to a certain conclusion. For example, let us say that a car manufacturer's Web site offers a good information architecture for users who are looking for the latest information about new products. However, if users want information about specific supplies (such as winter tires for the automobile), the information architecture is inadequate. The car manufacturer has fundamental

business motive behind this strategy. The car manufacturer wants users to contact a local dealer instead of using the Web to gain this type of information. In this example, an inadequate information architecture forces users to follow the preferred distribution channel and, in turn, helps the care manufacturer's business.

6.2 Key considerations and decision processes

Content accessibility on an intranet can have a direct impact on an organization's overall productivity. Organizations recognize the importance of information architecture, as both the primary prerequisite for designing and developing a new content management system and as the means to organize information in a way that is readily accessible to customer or employees.

One example is a call center that responds to customers using an intranet Portal to maintain contracts and to give information. For this call center, the range of time that is required for an agent to find the corresponding information and to fulfill the required steps ranges from under two minutes to four and a half minutes. Thus, if there are 80 attempts per day to find information and each requires two minutes more than necessary, this call center loses 160 minutes or 2.5 hours in productivity. Even without considering the costs that result from lost productivity, the customer satisfaction will decrease. When users cannot get the information that they need, the organization has a dissatisfied customer or, worse, a lost customer. Then, the effort to win back these customers is significant.

In addition to content accessibility, design flexibility becomes a key success factor for the information architecture of a Web site. One task is to conduct workshops with business stakeholders, including corporate communications and process area representatives. Given that the Web site will evolve as more business users provide input and, more importantly, contribute content, the site architecture can be expected to evolve as well. It is critical that the site framework be developed with flexibility for modification as a key goal.

Remember: In order to develop a good information architecture it is fundamental to understand what that information might be.

This section discusses topics that you should consider when planning the information architecture.

6.2.1 Understand the goals

When designing an information architecture, you first need to understand the business and user goals as well as the audience to whom the information is directed. Before you can look at structures, individual applications, or design aspects, you need to declare which are the business goals and the user goals per definition.

Web content management projects should be driven by defined business objectives, and the primary project goal needs to accomplish those requirements. When the project starts, the business objective is often exposed to discussion as more users provide input, because content providers have a better understanding of the information that the Web site provides. However, this understanding can mislead content providers into believing that they also have a better understanding of the business needs. You need a clear communication from the business constituency to monitor the motivation of the core team and the joint content provider. Unfortunately, Web content management projects often get delayed because of misconceptions between the core team, the content providers, and the users.

Before starting a Web site project, it is imperative that everyone involved understands the defined business goals and the user goals of the project. The challenge is to involve content owners in the defining process but to remain focused on the essentials. The user goals should be related to the users of your site (internal or external) and what experience you want them to achieve when visiting the site.

These two sets of objectives (business goals and user goals) need to be weighed against what is achievable in terms of budget, scope, and other limitations. When the organization understands both goals, there is a much higher chance that the content management project will be successful.

6.2.2 Understand your audience

The project effort will only show a return on investment if the audience can be reached with the business prospective. Any failure made at this point is significant for the project. First, you need to understand the demographics of the audience and why they are interested in visiting the site. What is the business prospective and does this equal the audience view of the business? Users do not have the same tolerance with Web information that they have for printed or broadcasted information. If you cannot meet your audience's needs, you will rarely get a second chance, especially when facing an external audience. In many cases, even employees will revoke an information source that is ineffective.

As stated in the beginning of this chapter, the only reason to share information about a Web site is to make it available for users. The Web site experience is the one experience that is most appreciated by the audience. The usability of the Web site is an essential subject. Usability is required to make sure that the audience can complete the tasks. Usability can only be achieved by meeting the needs and expectations of the target audience.

As a recommended best practice, it is very important to involve users during the process of designing the information architecture. Doing so helps to confirm the usability of the Web site and, in turn, to confirm an effective underlying information architecture. Unfortunately, in many projects, only content owners with a deep knowledge of the content are involved in the design process. While the content owners know the content very well, it is important to involve less experienced users to confirm that the navigation structure is intuitive.

A good approach to understanding the target audience is to gather facts about them. The more that you can gather characteristics that are based on real facts, the closer you can get to finding gaps in the understanding. At times, you might need to classify the audience into groups with individual needs in order to better understand the needs. You might also need to consider cultural, geographic, age, or technical differences. Some audiences need individualisms to have a positive Web site experience. What a specific user might consider appropriate and inviting can be very different from other users. Often, experiences from other Web sites can influence the users of your Web site. A good Web content management product offers the flexibility to set user expectations. Using traffic analysis can help to understand the interests and proceedings of users.

A Web site with a good understanding of the audience offers content in which the user is interested in a way that is intuitive and convenient.

6.2.3 Consider objectives and key success factors

For your Web content management product to be a success, you should consider and define the key success factors that will determine whether the Web site is meeting the objectives. Additionally, you also need to understand how you can measure the key success factors. Take note that higher Web site traffic is not necessarily an indication of a good Web site. For example, repeat visits can only be counted as a success indicator if the visits lead to a positive user experience or, in the case of a commercial Web site, to increase business. There are numerous mechanisms to measure the success of a Web site, including a simple feedback form, an analysis of complex statistics, or using custom-made measuring criteria.

Objectives and key success indicators might include the following:

- Avoid underestimating the role of the information architecture.
- Make a distinction between the business goals and the user goals.
- Understand the business goals and the user goals.
- Organize information in categories and levels.
- Ensure that the information architecture provides the following:
  - Intuitive navigation
  - Consistent structure
  - User friendly navigation, for both the first-time visitor and the frequent user, in a structure that serves the wide scope of various content
- Provide for flexibility in the design as the information architecture evolves. The site framework should also be flexible and easy to modify.
- Ensure clear communication to those involved with the project. Involve content owners in the defining process but remain focused on the essentials.
- Avoid using input from content owners who have an everyday comprehension of the content.
- Avoid having a single person or department to overlook all the content.
- Set one person as the decision maker.
- Ensure that the content is accessible.
- Ensure a good knowledge of the audience's Web site experience.
- Make a distinction between different types of users.
- Set a realistic time frame.

6.2.4 Find the `killer app`

One way to ensure use of the Web site is to identify at least one *killer app* — the one application or feature that everyone is likely to use frequently — and make that application or feature available only on the Web site. Most often an application or a particular type of information can be identified easily. Then, you can make this application easily accessible within the information architecture.

6.2.5 Consider a user-centric design

When you have determined the business goals and defined the audience for your Web site, it is time to start thinking about how visitors will use your site and how the design can ensure a high usability. *User-centric design* (UCD) ensures that the objectives of users are taken into account when designing the site. The

Archived

characteristic of a UCD approach places the user at the center of all design decisions when building a Web site. UCD also includes design in the manner of structure and information architecture.

One common complaint from users is that they cannot find what they need on the intranet. This does not mean necessarily that the site is designed poorly, but that the spectrum of users is comprised of linear and non-linear thinkers. Linear and non-linear approaches, as well as combinations of the two ends, are equally valid.

To accommodate the broadest community of users, IBM recommends that you develop three discrete design directions, focusing on page layout and navigation. Those design directions should then be presented to a focus group comprised of a cross-section of users. Feedback from those users provides very specific direction into how you should design the site and what content you should highlight. A facilitator should be engaged to conduct the session (it could be a qualified person within the content owner), and the session should be scripted with specific questions asked of the group.

A Web site with a high score in terms of usability would include the following characteristics:

- Information is easy to find.
- The site is simple to use.
- Pages open quickly.
- Creating and maintaining content is a painless process.
- The site is adaptable to changes.
- The site must be secure (where applicable), stable, and scalable for both the content and the user population.

As discussed in 6.2.2, “Understand your audience” on page 217, you might need to classify the audience into groups with individual needs. You can divide these different categories of users at least into two groups:

- First-time users
- Professional users

The AO Foundation example (discussed in 6.1.1, “Base example of a good information architecture” on page 201) shows a surgical management process that is used to offer specific AO Foundation knowledge that answers specific clinical questions. Two different UCDs help to support the users with the required information. The fist UCD provides intuitive, step-by-step decision-making support to lead surgeons (as shown in Figure 6-16 on page 221). This example can be adopted for any decision process in any organization.

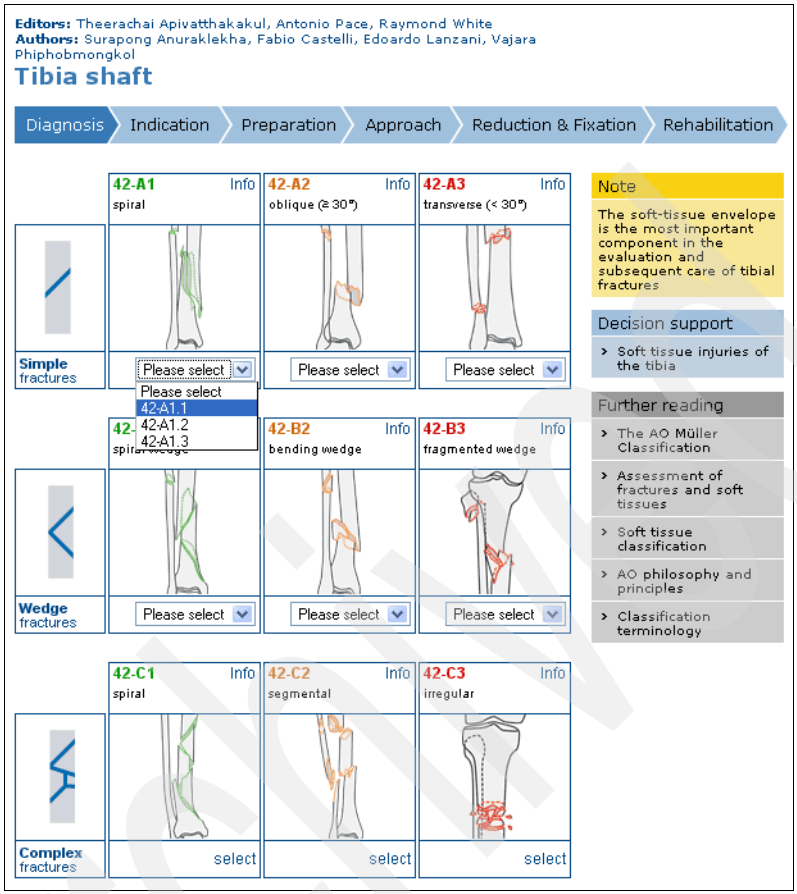


Figure 6-16 AO Foundation intuitive, step-by-step decision-making support

The second UCD is the AO Quickfinder (Figure 6-17 on page 222) for experienced surgeons who do not require the decision-making support and prefer to move directly to the specific clinical information that they need.

Archived

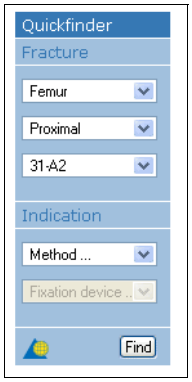


Figure 6-17 AO Quickfinder

You should also include a second involved target group when it comes to UCD. Consider persons following who will work with the Web content management system:

- Content authors (including content creators, approvers, and so forth)
- Site developers and site administrators

The success of an IBM Workplace Web Content Management project also depends on the usability for the every day editor's work. Their environment should be intuitive and consistently structured as well.

6.2.6 Define use case scenarios

We discussed the process of defining the audience's expectations at a Web site. In the next step, you define how users achieve the information. In order to get a first approach, it is good to define a number of key *use case scenarios* that represent the way that you expect users to interact with the Web site. You should define these use case scenarios from a user prospective in order to deliver an authentic scenario. They also need to include information about the involved context. The chosen use cases should include exceptional use cases in order to avoid the user getting stuck. The amount of detail that each user needs depends on the specific objectives of the Web site and the additional requirements that you have identified during the project.

The goal is to identify and build a picture of common tasks that users are willing to complete and how you anticipate the user completing these tasks. This iterative process starts with research about the audience, continues into the high-level design of the site, and is followed by a more detailed level of design that includes the structure.

Archived

6.2.7 Plan the site framework and site areas

As described earlier in this chapter, the site framework is the site map of a Web site. The site navigator (that is, a navigator component) can be generated from the site framework. The site framework is accessed through the Web site navigation. As an example of how to plan the site framework and site areas, we use the virtual River Bend Coffee and Tea Company Web site.

The information architecture analysis consists of two parts:

- A top-down approach investigates the content to be displayed on the site, plus the needs of the audience that is accessing the site. This approach helps to formulate the scope of the site and supports our understanding of how the overall site will be structured.
- A bottom-up assessment is performed. This starts from the most basic level of the Web site — the content page. This process determines what metadata, including categories and keywords, will be used to profile documents.

The goal of this analysis is to determine the structure of the site and, consequently, the navigation of the site. During this process, the following core facets of the information architecture are developed:

- Site framework
- Category hierarchy
- Document type hierarchy

Site framework

There are two ways of organizing content on a Web site: hierarchical and cross linking.

- Hierarchical organization

  Hierarchical organization of content is represented in Figure 6-18 on page 224. Each piece of content has a physical location at a level in the site. Because content is usually accessed from a single entry point (for example, the home page), hierarchical organization of content is particularly well suited to Web sites.

  Hierarchically structured content works best if you have well organized content that can be easily sorted into different levels on your site. Most visitors to a Web site understand the concept of hierarchies, such as chapters in a book. So, this is a common navigational aid. It provides at a glance an idea of the site's structure. This type of content organization is best suited to a less technical audience.

Archived

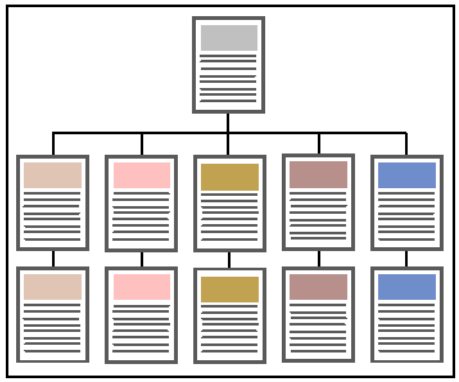


Figure 6-18 Hierarchical organization of content

- Cross linking

  Cross linking content is illustrated in Figure 6-19. With cross linking, any page can be linked to any other page on the site, regardless of its location within a hierarchy. This type of content organization works well in Web sites that enclose many hierarchal levels and allows content that is located on a deeper level within the Web site to be presented at a higher level.

  Cross linked content is well suited to a more experienced user and is intuitive to use only if the linking follows a content context that the audience easily understands. Frequently, only the back button of the browser allows the user to revisit pages.

Figure 6-19 cross linking content

In traditional Web sites, the site framework is an artificial concept based on links between pages. In IBM Workplace Web Content Management, the site

Archived

framework is defined first, and the links between the pages are generated dynamically. The different sections of a Web site are defined in site areas, and content is grouped under different site areas. Navigators are based on the site framework. By clicking a navigator, a user can access all content in a specific section. Menus display lists of content that can be located in any section. Menus usually show content that has been assigned the same category (although there are many search criteria that menus can use, including site framework and document type).

Category taxonomy

The next stage is to create a hierarchical classification system used when displaying content. *Category taxonomy* is simply the classification of content into categories. When you have created this taxonomy, you can use it to apply a set of standard categories to content. The category taxonomy defines the category attributes that can be assigned to content that is located anywhere in the site. By referencing different pages of content that have been assigned the same category, content from different locations can be grouped on one page.

A *category* refers to the subject matter of content or what your content is about (for example *news* or even the product type *black tea*).

When creating taxonomy of categories, the main goal is making it easy for the Web site users and search engines to find specific content. An added benefit is that an organization can begin to apply standard practices around the classification of content inside the organization.

How many levels to create?

If you find that you have so many documents covering various topics that when searching for a document based on a category, an exhaustive list is returned, it is time to think about creating new categories. You might want to assign a category to do the following:

- Make it easier and more effective to find the information for which you are looking.
- Enable more effective searching on content (either within the site or by search engines).

There are two main standards groups doing work in the area of categorization:

- The Dublin Core Metadata Initiative includes specifications of the terms to be used when creating metadata. You can find information at:

  http://dublincore.org

Archived

- The World Wide Web Consortium's Resource Description Framework (RDF) provides recommendations about the way to represent resources on the Web. You can find information at:

  http://www.w3.org

The initiatives of these two organizations provide companies with the ability to standardize the metadata tags that they used on their site. These initiatives provide standards for terminology to be used in category taxonomy. However, they provide little assistance in the creation of the taxonomy itself.

Creating a category taxonomy

Categories are grouped within *taxonomies*. Categories are also grouped into *parents* and *children* within a taxonomy. When building a hierarchy of taxonomies and categories, it is important to consider how a menu will use your categories in a search, because menus search both upwards and downwards within groups of categories, as follows:

- If you base a menu on a top-level category, all content profiled with categories belonging to that top-level category and their children appears in the menu.
- If you base a menu on a parent-level category, all content profiled with the parent-level category, its children, or its parents appear in the menu. If you base a menu on a child-level category, all content profiled with the child category or its parents is returned by the menu.

It is recommended that, in most cases, you use only child-level categories to profile content to allow more control over what displays in menus.

Taxonomies cannot be used in Web content management menu searches. If you would like to have a menu return results based on content that is profiled with any category in taxonomy, you should create a single, top-level category and base the menu on the top-level category.

Archived

For our fictitious Web site for the River Bend Coffee and Tea Company, we organized the content by the five sections Products, Company, Locations, News and Contact (as shown in Figure 6-20). Our reference implementation site structure reflects this organization.

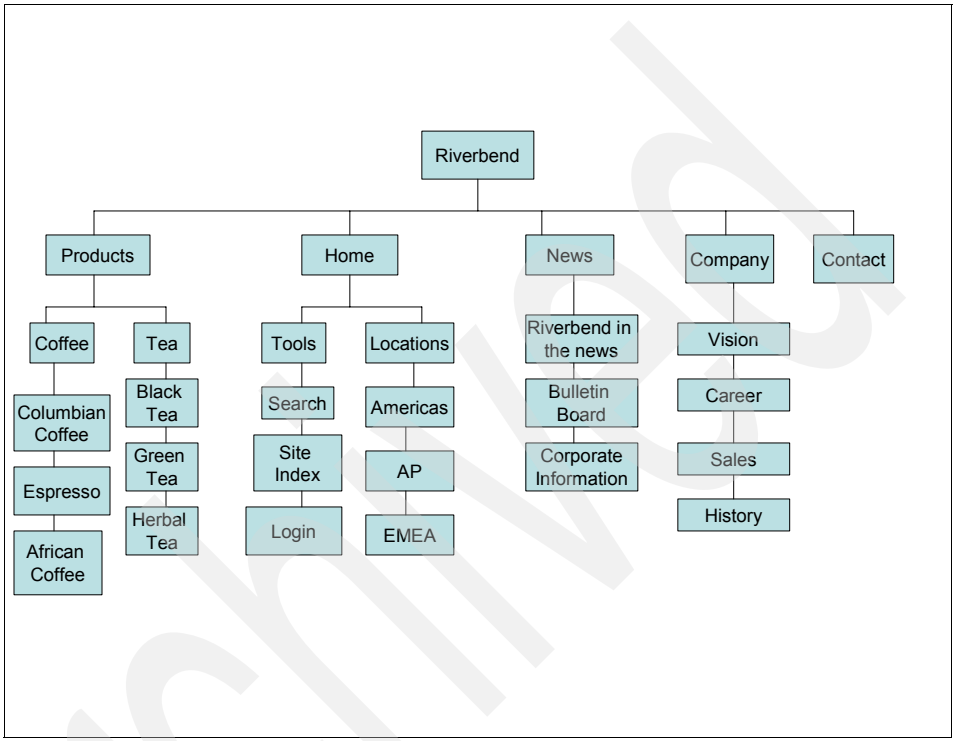


Figure 6-20 The River Bend Coffee and Tea Company navigation

The secondary site areas allow specifying what the user is looking for in this area. The secondary area for Products include *coffee* or *tea*. The separation is very obvious, and users can comprehend intuitively and exactly what they can expect to find here.

Document type hierarchy

With IBM Workplace Web Content Management, content is created using authoring templates. These templates are forms that contain content and can be grouped into different document types. When it comes to defining the required document types, it is common to hear content owners requesting a document type for every propose. For example, the sales team wants a document type for products, human resources wants document types for career information, and marketing sees a need for a document type for news and company information.

Archived

From a technical perspective, these requirements fit in fewer authoring templates because they are quite similar essentially.

Instead of creating a new template for every document type on the Web site, templates can differ by page design, such as page layout, page style, and components. Then, you can apply these templates to the many content document types that you create so that the content propose is separated from its layout, making content and presentation flexible. Proper construction of the shell site ensures maximum re-use, flexibility, and performance.

Formulating ideas about other components to be used in the Web site, such as graphics, menus, navigators, and so on, is also helpful. You can find details on designing the Web site and implementing it in Chapter 5, “Key concepts and terminology” on page 141.

6.2.8 Metadata, keywords, and categories

When content is created, it is crucial that you define descriptive information for that content so that it can be categorized and searched. descriptive information is called *metadata*, as defined by the W3C (`http://www.w3.org/Metadata/`) . Metadata is machine understandable information for the Web. It is required to include metadata as a part of the Web that contains *information about information*. Metadata include labeling, cataloging, and descriptive information that is structured in such a way that allows Web pages to be properly searched and processed in particular *by computers*. Metadata is most likely keywords about the content of digital material in the form of labels that are only visible in the HTML code that builds the Web page. The metadata should match with keywords that a user would associate with the content.

In our example, we can plan to generate metadata for IBM Workplace Web Content Management pages by specifying standard metadata fields. The search supports the metadata fields described in Table 6-1.

Table 6-1 Metadata fields supported

Name	The title of the content item.
Description	A description of the content item.
Author	The authors of the content item.
keywords	Keywords that are associated with the content item.
Subject	Taxonomy and category information associated with the content item.

This information must be created automatically for each Web Page.

> **Note:** WebSphere Portal Experts GmbH have realized a solution for the usage of metadata functions using IBM Workplace Web Content Management inside a WebSphere Portal (`http://www.wpexperts.com`). WebSphere Portal Experts Portlet service allows exchanging data between a Portlet and the Portal themes state service.

Refer to the search methods detailed in Chapter 10, “Search functionality for IBM Workplace Web Content Management” on page 429.

6.2.9 Determining the components of the Web site

To give an overview of components of the Web site, we look at the River Bend Coffee and Tea Company that we use as the example site throughout this book. Figure 6-21 shows an overview of the site.

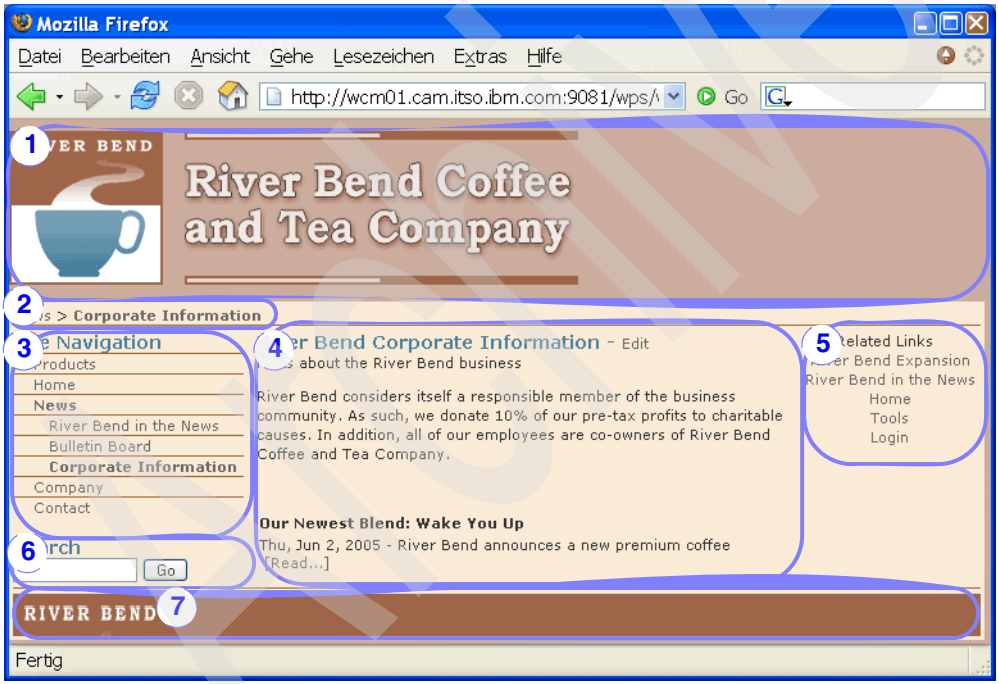


Figure 6-21 River Bend Coffee and Tea Company page overview

Archived

The River Bend Coffee and Tea Company Web site is structured in seven components as follows:

1. The Company Logo as the head banner.
2. The Breadcrumb navigation displays the current page's context within the site structure. The term *breadcrumb* refers to leaving a trail in order to find your way back to where you started. This popular approach makes it obvious to users which path they took and allows them to move between these navigation levels and to understand the information structure.
3. The Navigation covers five top levels of site areas or containers for content that are accessible directly from the home page — Products, Company, Locations, News, and Contact. Figure 6-21 on page 229 shows the expanded News level.
4. The Content Area displays the selected content.
5. The Site Menu delivers additional information about the currently selected content.
6. The Search runs over the whole Web content and delivers a result list inside Content area.
7. The Tool bar at the bottom of the Web page includes a company logo, leaving open the option to expand the functionality on the Web site.

The chosen navigation contains all level of the River Bend Coffee and Tea Company site. Thus, the audience is not required to jump to a second navigation while researching the Web site. The navigator expands downward and can be used to support further site areas.

6.2.10 Using workflow and syndication

One of the advantages of using an IBM Workplace Web Content Management system is to dispense the content creation process over different locations or within different security levels. With multiple locations (servers) in a typical IBM Workplace Web Content Management environment, it is necessary to have a mechanism that can synchronize content between the multiple locations. In addition, a typical requirement has the content pass through some formal approval process before it can be viewed on the Web site. IBM Workplace Web Content Management provides two complementary capabilities for these issues — workflow and syndication.

Creating a workflow

Workflows are required within IBM Workplace Web Content Management to approve and to authorize content that is published in the Web site. You need to create at least two stages (Draft and Published) inside a workflow. In the simplest

case, the content creator is also the person who publishes the content. This configuration supports a rapid content creation process but is contrary to the motive of workflow. However, this is the most often use of workflows inside Web content management. Organizations find it favorable to accelerate the content creation process by skipping the workflow.

Important: The process of content creation is always underestimated!

There is a high danger in underestimating the value of the workflow process. The workflow approves the content quality and assures that content is published in an organization policy manner. Without a workflow, the consequences are that content can be published in a Web site that is identical with printed statements. This is true for internal and external published content.

A recommended workflow contains the stages Draft, Approval, Published, and Archived. Every stage is logged inside a content history that makes traceability easier. IBM Workplace Web Content Management offers various options to configure more dedicated workflows to serve further needs. An additional option to define workflows not supported within IBM Workplace Web Content Management is to use its API and IBM Process Choreographer. For more information, see Chapter 12, “Working with the application programming interface” on page 469.

6.2.11 Using syndication

Syndication is the method used by an IBM Workplace Web Content Management location to exchange (replicate) data from one IBM Workplace Web Content Management location to another. Unlike workflow, syndication is not involved in the process of approving content. Syndication is only responsible for replication of IBM Workplace Web Content Management assets across multiple locations.

When creating a syndication process, IBM Workplace Web Content Management allows you two choices for the granularity of items to be syndicated:

- All Items

  When configured with this setting, all elements of the IBM Workplace Web Content Management location are replicated to the other location, including draft and expired content. Typically, authoring and development locations need access to all types of content for testing purposes. This setting is common when syndicating between distributed authoring locations or between an authoring location and a development location.

- All Live Items

  As opposed to the *All Items* setting, this option replicates only live content. Live content is defined as content that has been approved via workflow (live) but that is not yet past its expiration date. Typically, a production location only receives live data, so this setting is common when syndicating from authoring to staging or staging to production.

Tip: Syndication always syndicates copies of your technical assets (for example, presentation templates, authoring templates, library components, and so forth). By default, there is no setting to stop this syndication. You can configure IBM Workplace Web Content Management to workflow these items, at which point they would follow the same rules as content.

6.3 Information design

A critical component of information architecture is its design, which addresses content accessibility as it pertains to the site interface. Page layout, links, menus, navigators, taxonomy, site breadcrumbs, search, identity standards, graphics, and site metaphor are all components of Web information design. Different users have different preferred means for accessing information. Thus, an effective information design provides users with multiple means for accessing information. The most common means are:

- Direct feature links
- Embedded hyperlinks
- Site navigators
- Site menus
- Search

All means are valid, and we recommend that you leverage all site components from IBM Workplace Web Content Management when designing and building a site to enhance user experience. To provide a relevant context for these components and to illustrate specific examples, we draw upon examples from River Bend, IBM developerWorks, AO Foundation, WebSphere Portal Experts, and Ascendant Technology sites. We use these throughout this book, because we believe that they represent good information architecture and good site design. Other than River Bend, which is a fictional company, you can access these company Web sites as follows:

- IBM developerWorks (`http://www.ibm.com/developerworks`)
- AO Foundation (`http://www.aofoundation.org`)
- WebSphere Portal Experts GmbH (`http://www.wpexperts.com`)
- Ascendant Technology (`http://www.atech.com`)

6.3.1 Using presentation templates

One of the features of IBM Workplace Web Content Management is that you can separate the creation of content from the presentation of content using *presentation template*s. When using presentation templates, the content creator is not responsible for the page design that will ultimately display their content document. The presentation template is determined by a relationship in the site between a page design and content authoring template. It is recommended that you avoid a large number of slightly different presentation templates and instead, use one that covers all authoring template features.

It is possible to pair different page designs (presentation templates) with one authoring template in different site areas. This design means that one piece of content can be displayed in multiple ways depending on the site area that is currently displaying — called *multipublishing*. Different uses of multipublishing can be helpful. You should create one content that includes a full set of information, for example, product information. This content can contain confidential material that is published in an intranet or extranet. If the same content is taken and published in the Internet, different presentation templates allow you not only to have a different design but also to share a subset of the original content securely. You can create content with the same type of authoring template but publish it with different presentation templates, depending on the department, brand, location, or language.

To increase the reuse and to decrease the maintenance effort, you can use components. Presentation templates can contain different kinds of components that are used depending on the site area on which the content is published. Thus, when news is separated into categories, depending on the site area, a different title logo (component) is chosen instead of creating a presentation template for every news category. Components are very helpful in customizing presentation templates but should be used reasonably.

Archived

6.3.2 Using site navigators

The information design of navigators is a profound part of the user's Web experience. Some users prefer to drill down through navigation levels with the site structure being as big a part of the user experience as the content itself. When it becomes necessary to include a high rank of navigation levels, it is crucial to choose an information design that stays straightforward and intuitive. A very powerful feature is highlighting the selected navigation path. Users easily get confused when the navigation does not follow a logical structure or changes its design.

Related to the usability considerations, you should also consider the performance of navigation. To save space or because it might seem more high tech, navigators are used that enfold all available navigation levels by clicking or mouse over. This kind of navigation is accomplish often through complex handling and cannot be easily overlooked when Web accessibility is a big issue (`http://www.w3.org/WAI/`). Public Web sites should support Web accessibility, which is at times required by law.

Because IBM Workplace Web Content Management is a dynamic process that lives over years, navigation should allow the site to expand dynamically as well.

Note: Navigators are not menus. *Menus* are a list of hyperlinks that take you to specific pages. *Navigators* can also be hyperlinks that can take you to specific pages, but navigators are organized differently. Navigators present the logical arrangement of a Web site, while menus are a list of related Web pages that might appear anywhere in your site and follow search criteria.

Archived

The AO Foundation navigation in the site area *Case of the month* comprises six levels in a in a very intuitive usage, as shown in Figure 6-22.

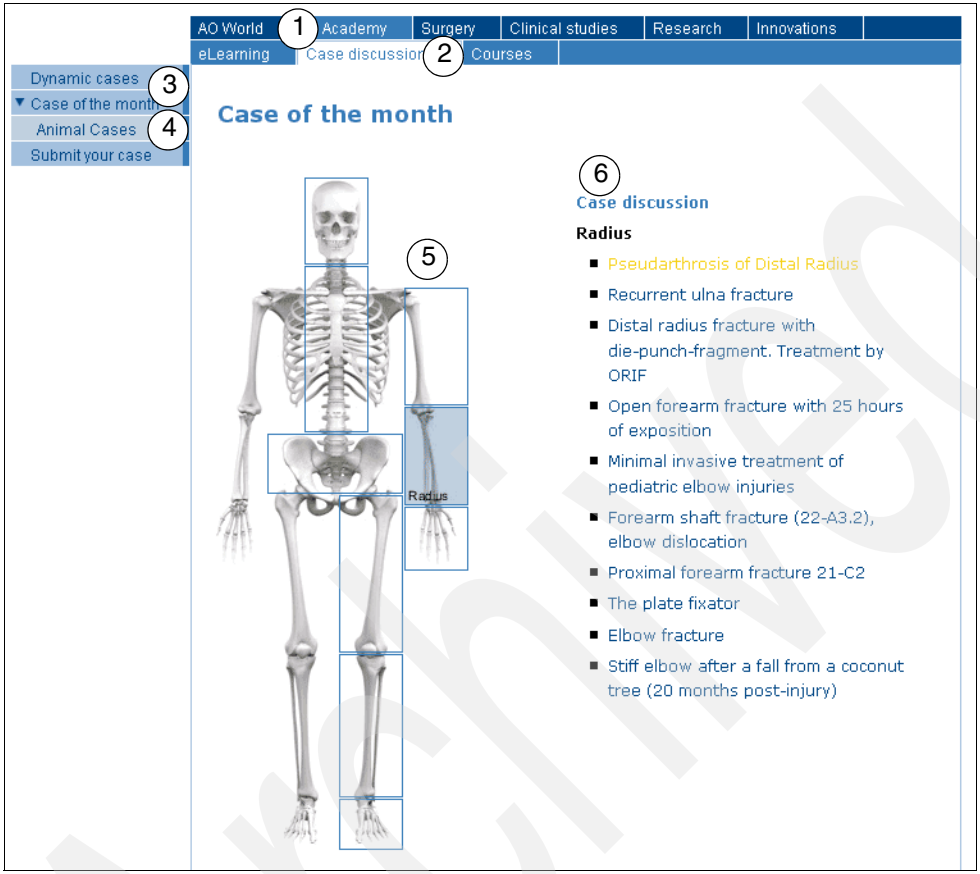


Figure 6-22 AO Foundation six-level navigation

The first and second navigation levels are located in the upper horizontal navigation where the selected topics are highlighted. Instead of overloading the overall navigation with more text in the fifth-level navigation, it is structured within a human skeleton. This kind of navigation is intuitive when the object in the graphic is familiar to the audience.

Archived

IBM developerWorks also splits their navigation into different parts. As shown in Figure 6-23, the first-level navigation is horizontal on the top of the page similar to a toolbar. The main navigation of the site area is the vertical-level navigation that expands downward. This navigation includes level two and three of the Web site area.

Figure 6-23 IBM developerWorks navigation

When the second level is selected, the user gets a content navigation that offers more detailed information about the navigation topics (Figure 6-24). This kind of information design gives the audience a more detailed navigation experience but also leads to a change in the navigation design. For IBM developerWorks, a technical audience is expected, and that audience is not confused by this design break. However, other audiences might prefer to be guided through the Web site using a single navigation.

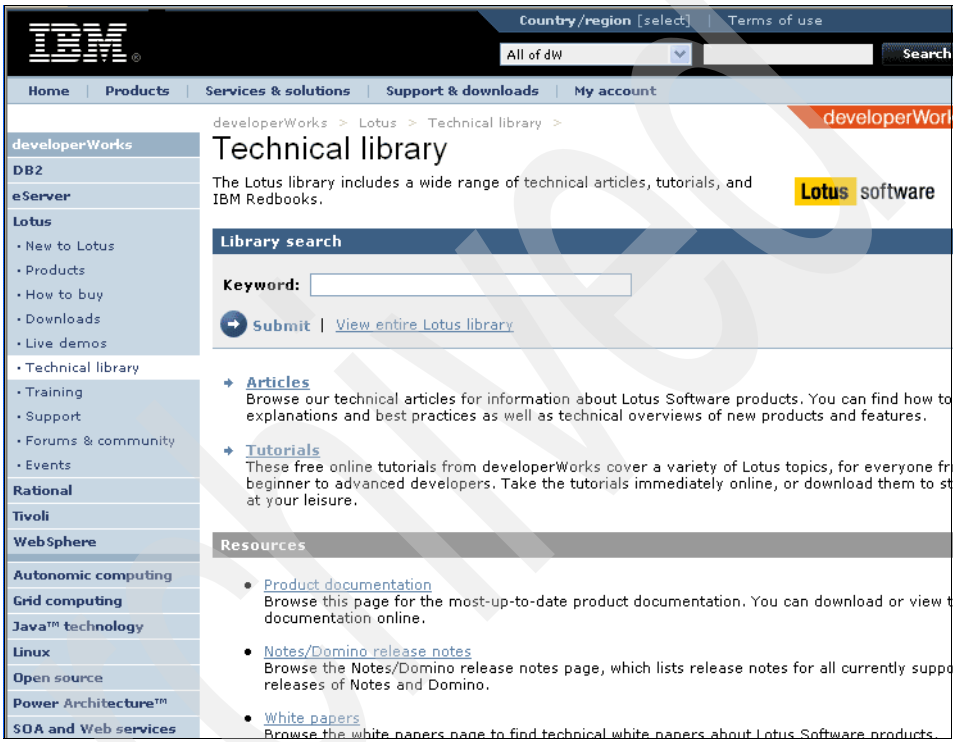


Figure 6-24 IBM developerWorks fourth-level navigation

Archived

6.3.3 Using menus

A *menu* is ultimately a search mechanism that groups together lists and related documents based on specific criteria, such as categories, site areas, or keywords for display and access to users. Figure 6-25 displays links as images, icons with text, links with a summary, or many other combinations. These menus combine elements based on documents from more than one site area, where it is not possible to use site navigators. To users, menus and navigators provide the same functionality in that they allow a user to navigate around the site or to locate particular content.

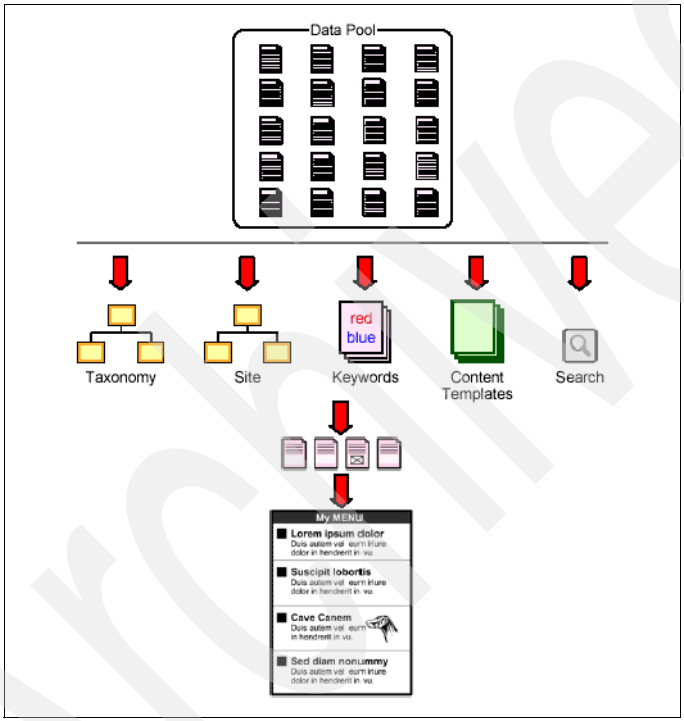


Figure 6-25 Menu construction from various selection criteria

Archived

WebSphere Portal Experts offers menus for all its content pages, as shown in Figure 6-26). Thus, menus are embedded as rich text components and are managed centrally. This solution enables WebSphere Portal Experts to re-use the same links wherever they are applicable, thus limiting the maintenance effort significantly.

Figure 6-26 WebSphere Portal Experts Menu

AO Foundation also integrates menus to assist the decision support of surgeons by offering *Further readings* (Figure 6-27). This content is stored in a separated area and can be matched with related information. Instead of pointing the audience to a book requiring additional effort, the content of those books is included where it is needed to give the best user experience.

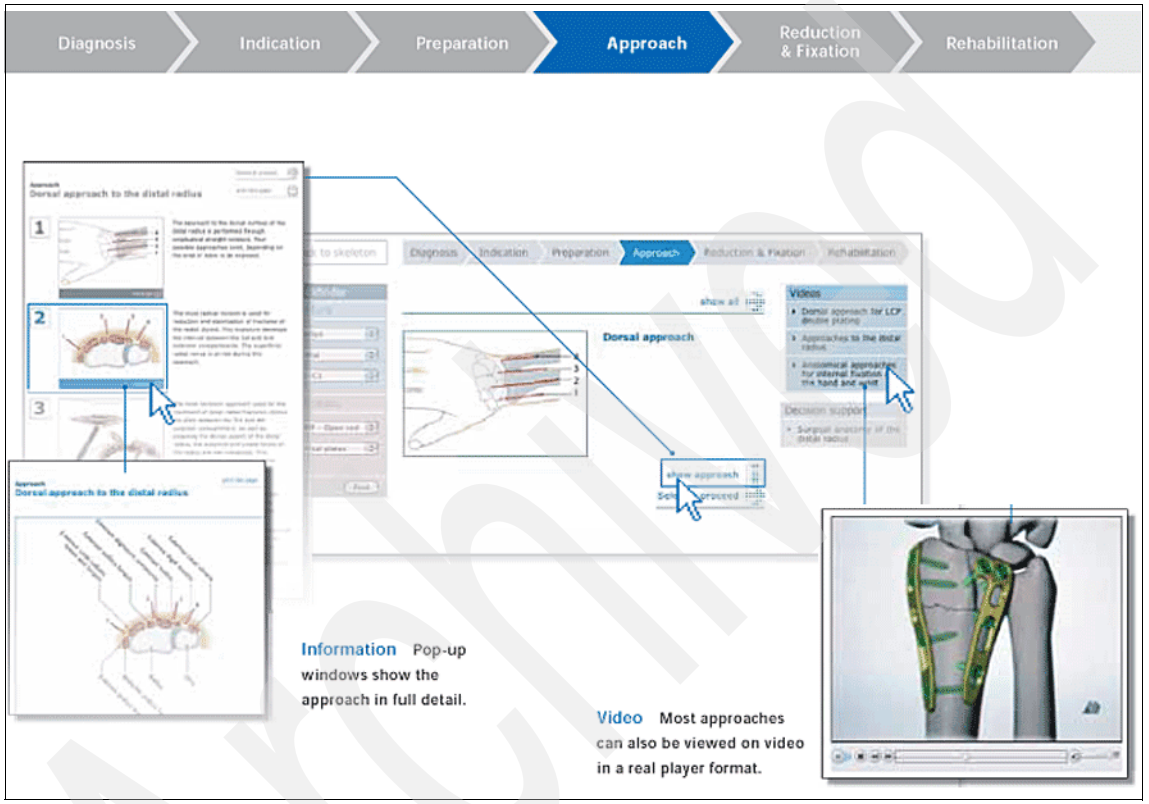


Figure 6-27 AO Foundation integrated menus

Archived

6.3.4 Using direct feature links

Of all the methods to access content, direct feature links is the most direct and provides the most premium. This method enables one-click access from a top-level site location, which is almost always the home page, to any content that is embedded a few layers down in the site structure. These types of links are reserved for the most timely and relevant news or information. However, they can also be used to provide additional information that is related to the current content.

Through the use of menus, and leveraging categories, parameters can be set to bring the timeliest content to the home page. In addition, individual pieces of content can be categorized, or a category can be associated with a specific content template so that all content created with that template has the specified category. Content can have one or more categories. These categories then can be presented dynamically by direct feature links such as News, Products, Events, or Promotions.

Ascendant Technology provides direct feature links to all their content pages, illustrated in Figure 6-28 on page 242. These direct feature links are embedded as rich text components and are central managed. This solution enables Ascendant Technology to re-use the same links wherever they are applicable.

Archived

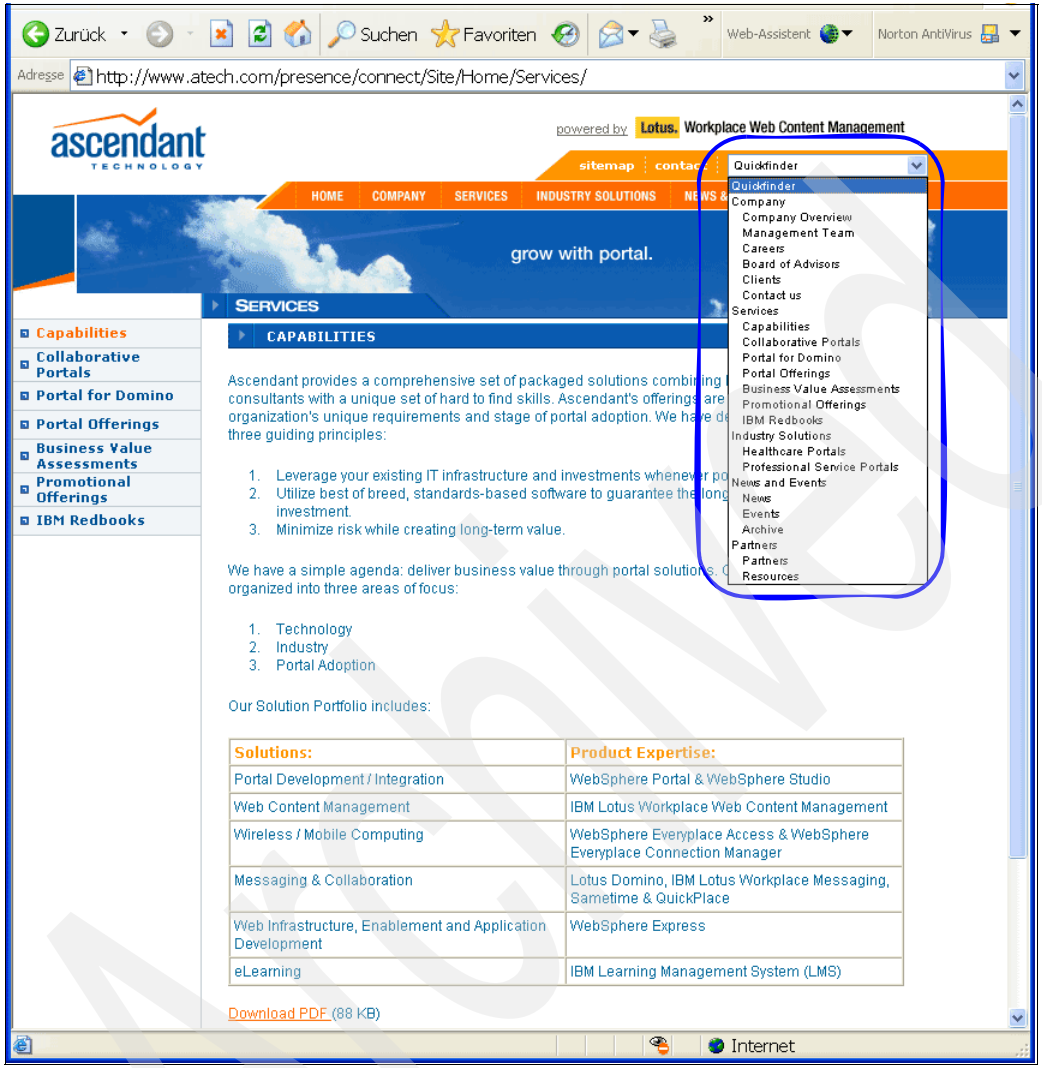


Figure 6-28 Ascendant Technology direct feature links

We recommend that every site include the Search, Contact, and About us direct feature links to provide the user with a good Web experience.

6.3.5 Using embedded hyperlinks

Embedded text hyperlinks provide users with one-click access to content that i s related to the article that they are reading but that is location perhaps in a different area of the Web site. The only way for users to return from embedded hyperlinks is to use the browser Back button. Navigators and breadcrumbs cannot do not work here. When pages include links to pages outside of the Web site, you should replace the current Web site when contacting the new link or open the new site in a separate window. Often, embedded hyperlinks are used to define a word or phrase that is mentioned (much like a cross reference in a book), as shown in Figure 6-29.

Figure 6-29 WebSphere Portal Experts embedded hyperlinks

Another usage of embedded hyperlinks is in WebSphere Portals where they can be used to reference corresponding content or applications, depending or related information. This WebSphere Portal feature is called *click-to-action*.
Click-to-action capability lets you send information from one portlet to another so that you do not have to enter cross-portlet information manually. This capability leverages the way that Web content and Web applications interact and leads to more powerful dynamic workplaces. For example, if a user clicks on a specific topic in a Topic List portlet, click-to-action allows an adjacent portlet to limit the postings shown to just those in the specified topic, based on the criteria selected in the Topics portlet.

6.3.6 Using search

The search of a Web site is one of the most used features and is particularly useful to users who are pressed for time and who do not care to invest time in learning about the site structure. Visitors expect the search to return exactly what they are looking for, especially when they know that the information is there. Because of this high expectation, search results need to be high quality. To return effective search results, it is critical that content categories and keywords are defined and applied to each page prior to publication. See Chapter 10, “Search functionality for IBM Workplace Web Content Management” on page 429 for information related to the search setup and configuration.

So, how should a search work from an information design perspective? There are two options possible. First, the search function should be a simple search that is intuitive to use (Figure 6-30). The user can enter words in an input field, and the results deliver all contents that include the searched parameters, sorted by relevance. Relevance does mean it starts with the content that includes all words searched for and continues with single words found.

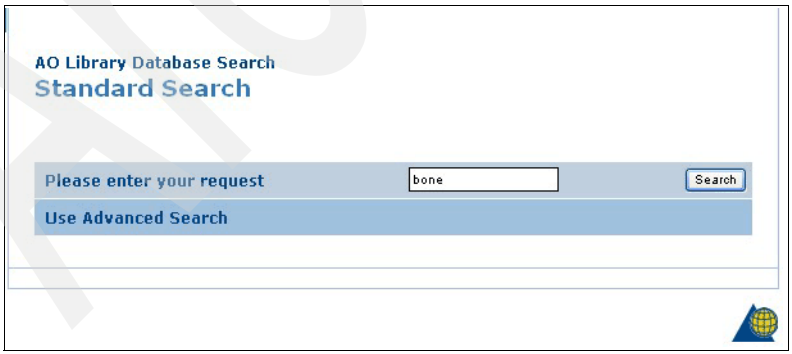


Figure 6-30 AO Foundation standard search

The second option includes a search function that offers an advanced search feature. The advanced search depends on exactly the needs of the offered content. Details could differ from a search combination linked by `AND`, `OR`, or `NOT` operations up to very specific needs of an application (Figure 6-31).

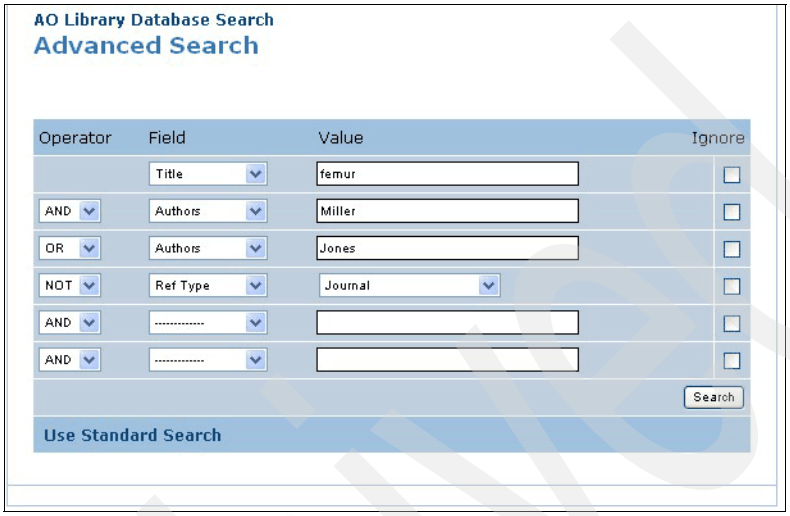


Figure 6-31 AO Foundation Advanced search

Archived

Searches can be run over not only the content of a Web site but also included databases, such as the literature database used on the AO Foundation Web site. AO Foundation took it one step further and gave the audience an experience that is rarely reached and included when searching the AO Members information from the LDAP. So, a search for a special person not only returns all content in which that person's name is included but also any details about that person (Figure 6-32). In addition, the personal information that is shown is more detailed if the search is started within the member area.

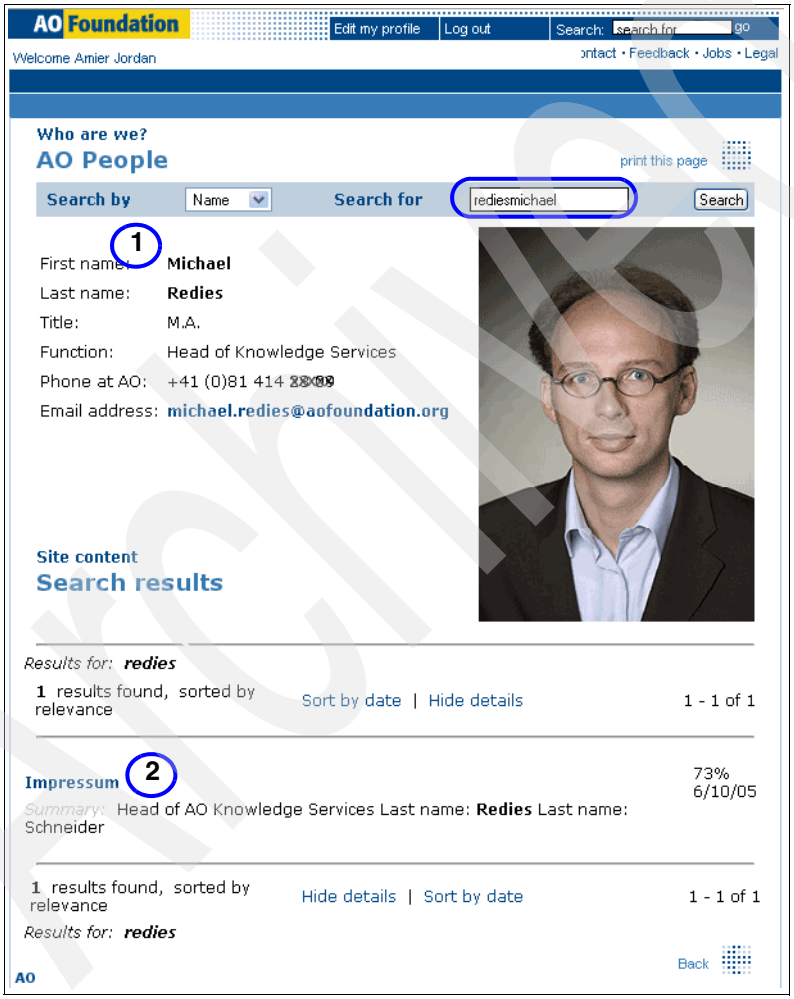


Figure 6-32 AO Foundation Search with content and LDAP results

Archived

However, what if the search returns hundreds of results? A good option is to offer users an option to search within the results of the first search. Technically, this means to use the first search and add a second search string to it (Figure 6-33).

Figure 6-33 AO Foundation search within results

An organization might also want to offer anonymous users different searches from authenticated members, where the included content or applications varies based on who is performing the search.

Search results can also have a different approach. Search results should generally offer a URL to the Web page that includes the cited content. To enhance the user's experience, a short abstract can also be part of the result. This abstract offers a specified number of words from the content or especially for the search result added content fields. In particular, when searching databases, results should use the advantage the structured content offers in an informative way. Details of the results can be saved for download or can be printed.

Archived

AO Foundation used this option to include their literature library of all existing written information, as shown in Figure 6-34.

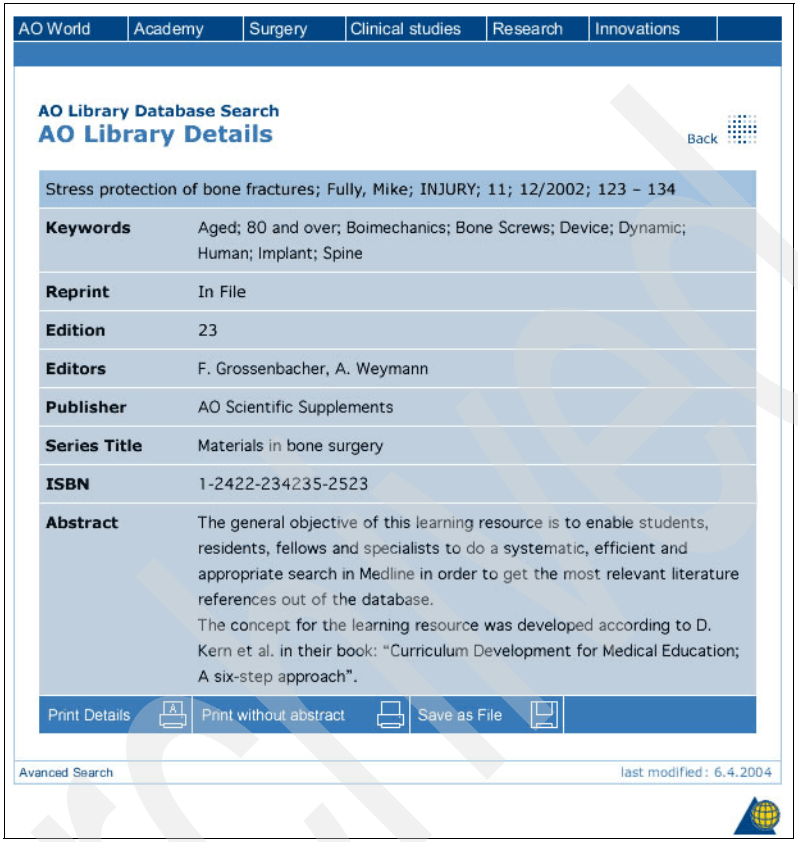


Figure 6-34 AO Foundation literature library details

There is one topic that is often forgotten when it comes to search results. If the Web site contains multilanguage content, would the visitor expect to find results in a language other then the language searched from when entering a keyword? The answer is most likely no. Language is also a search topic that should be taken into account.

6.4 Site acceptance

> **Note:** The value of an open communication about the IBM Workplace Web Content Management projects business values and key targets is often underestimated. These communications are crucial to the success of this type of project and the site acceptance.

An IBM Workplace Web Content Management project can only be successful if the persons involved in the project are motivated to give their part. Often, these kinds of projects involve for the first time a wide range of job roles within an organization. Other then many other IT or knowledge projects, a Web site — whether internal or external — most likely involves all parts of the organization. Organizations can underestimate the political, local, and cultural problems associated with a project of this type, and this underestimation can become the biggest challenge of the project. Therefore, it is crucial to communicate the reason and business value of the project and to get a wide acceptance of the project for it to be successful.

One challenge constantly voiced by technical teams is the lack of site use by many of organization's staff and the lack of content contribution by the business stakeholders. We recommend a few basic actions to help build support for the Web site.

6.4.1 Business stakeholders

Identify a set of business stakeholders, including the target group (not necessarily the designated intranet liaisons) to not only contribute to information architecture sessions but to provide feedback on general layout, design direction, and so forth. Ask these stakeholders to communicate within their groups, identify potential power users, and commit contributions of new content to the site. Consider leveraging the power users as beta testers of the site, both as users and content authors.

6.4.2 Design and layout approach

To accommodate the broadest community of users, we recommend developing three discrete design directions, focusing on page layout and navigation. Those design directions should then be presented to a focus group comprised of a cross-section of customer users. Feedback from those users provides specific direction into how the site should be designed and what content should be highlighted. A facilitator should be engaged to conduct the session (it could be a qualified person within the organization), and the session must be scripted with specific questions asked of the group.

6.4.3 Communications and enablement

As with any deployment of a new system, the organization should develop a communications plan to introduce the new Web site. If power users are identified within the process areas, they can help provide a level of enablement and support within their groups when the new site is launched.

6.4.4 Wireframe demonstration

The communication should contain a wireframe demonstration that illustrates the site navigation. In that demonstration, very simple action settings should be enabled for demonstrating the multiple ways of providing information. Not all information access methods need to be demonstrated. Open and agreed areas should be identified and marked. In meetings with business users, demonstrations such as this require little preparation time but are effective in communicating accessibility and usability concepts.

6.5 Workshop approach

Organizations planning an IBM Workplace Web Content Management project should conduct information architecture workshops with business representatives of each of the process areas over the first period. This section outlines some recommendations for preparing and conducting workshops.

6.5.1 Workshop preparation

Information architecture workshops require preparation in order to be effective. Prior to conducting any workshops, we recommend that organizations develop a test site with IBM Workplace Web Content Management to gain hands-on experience with the product and to develop an understanding of the capabilities of the tool. This training helps the team to connect user requirements or wishes (the *what*) with the capability of the tool to support them (the *how*). With that understanding, stakeholder requests that are not technically feasible can be identified immediately, so that expectations can be set correctly. Alternatives can be discussed on the spot.

After a workshop is scheduled with the stakeholders of a process area, the organizations team should thoroughly review that area of the site. Site areas and a sampling of specific pieces of content should be identified and documented. Separate documented notes by content type or site area but make no additional marking on them prior to the workshop.

Archived

To reach a high acceptance, we recommend a professional preparation. Because the workshop topics can get quite challenging, you do not want to spend any time discussing topics that are related to future topics. We recommend setting up the conference room for the workshop prior to its start (for example, draw a couple of page layout grids on a whiteboard and include some of the page elements that are persistent throughout the site.

6.5.2 The workshop

Unlike a workshop with the technical team, workshops with business stakeholders should avoid technical terminology or a discussion about how the IBM Workplace Web Content Management tool functions. A suggested approach includes:

- Introduce the goals and objectives of the workshop, and the overall IBM Workplace Web Content Management project.
- Identify basic terms (grid, banner, header, navigator, and menu) that will be used to produce a work product for this process area.
- Have the business stakeholders validate the site areas and content that was identified in the preparation.
- Use notes to group like content types, and place the notes on the whiteboard within the grid.
- Use a system of dots to identify site levels (one dot for the top, two for the next, and so forth).
- Capture the raw output of the workshop on digital camera.

6.5.3 Work products

The workshop outputs should include:

- A map of the process site area hierarchy, usually developed in Visio®.
- Wireframes of the page layouts, usually drilling down two to three levels and normally developed in PowerPoint®.
- A simple demonstration of page navigation with PowerPoint or simple HTML.

Archived

Archived

7

Building the River Bend Web site with IBM Workplace Web Content Management

This chapter describes step-by-step how to create a Web site using IBM Workplace Web Content Management. In this chapter, we refer to the site that we created for the River Bend Coffee and Tea Company as the *River Bend Web site*. The Web site contains content that describes the company's business and provides information about its products.

© Copyright IBM Corp. 2006. All rights reserved.

Note: The method that we use to build a Web site follows closely the approach that is used in the Lotus Education course *Developing Web Sites Using IBM Workplace Web Content Management 2.5*, which is offered in an eLearning format (course code LWWCM445) and in a classroom format (course code LWWCM440).

If you have already taken this course, then most likely you are familiar with the River Bend Web site and the techniques that are used to build the basic site structure. The course guides you through every step of building the Web site.

Alternatively, this chapter provides a detailed, step-by-step guide that goes beyond the scope of the Lotus Education course. It includes topics on Portal Document Manager integration and describes the basic approach for integrating a stand-alone site into a Portal.

Attention: While this chapter guides you through building a basic Web site, it assumes that you have also reviewed and understand the key concepts that are discussed in Chapter 5, “Key concepts and terminology” on page 141. We highly recommend that you are familiar with the contents of this chapter prior to building a Web site.

Tip: The resources used to build the sample Web site, including images, code samples, and stylesheets, are available for download from the following:

`ftp://www.redbooks.ibm.com/redbooks/SG246792/`

Archived

7.1 An overview of the scenario

For the purpose of providing a realistic business context to this chapter, we use the River Bend Coffee and Tea Company as the basis for the development scenario. River Bend Coffee and Tea Company, a subsidiary of WWCorp, is a fictitious company that uses IBM Workplace software. It operates a chain of 20 retail stores in 12 cities worldwide. In addition, the company runs an Internet-based retail operation, offers small-scale catering services, and has launched a certification program for employees and customers who wish to become skilled roast masters.

Figure 7-1 provides a brief overview of the site which we build in this chapter.

Figure 7-1 The River Bend Web site that we build in this chapter

Currently, the company uses IBM Workplace to:

- Provide messaging and e-mail services to its remote employees.
- Facilitate collaboration and communication among its geographically dispersed work force.
- Communicate and coordinate relationships with its vendors and customers.
- Manage, deliver, and track training to employees and customers.
- Support enterprise-wide document management, including regulatory compliance, corporate accountability, and information flow.

Archived

- Communicate with a sibling subsidiary, a retail organization that specializes in recorded music products.

> **Note:** The sample screen shots and tasks that we present throughout the chapter are intended to support the development scenario. The fictitious user names, documents, titles, and store locations are used solely to simulate the IBM Workplace environment and to demonstrate functions and features. They are not intended to depict real people, content, or company policies. Any likeness to real persons, content, or companies is coincidental and unintended.

7.2 Overview of the process

Throughout this chapter, we guide you through the process of building a sample Web site using IBM Workplace Web Content Management. Figure 7-2 provides an overview of the high-level steps in this process.

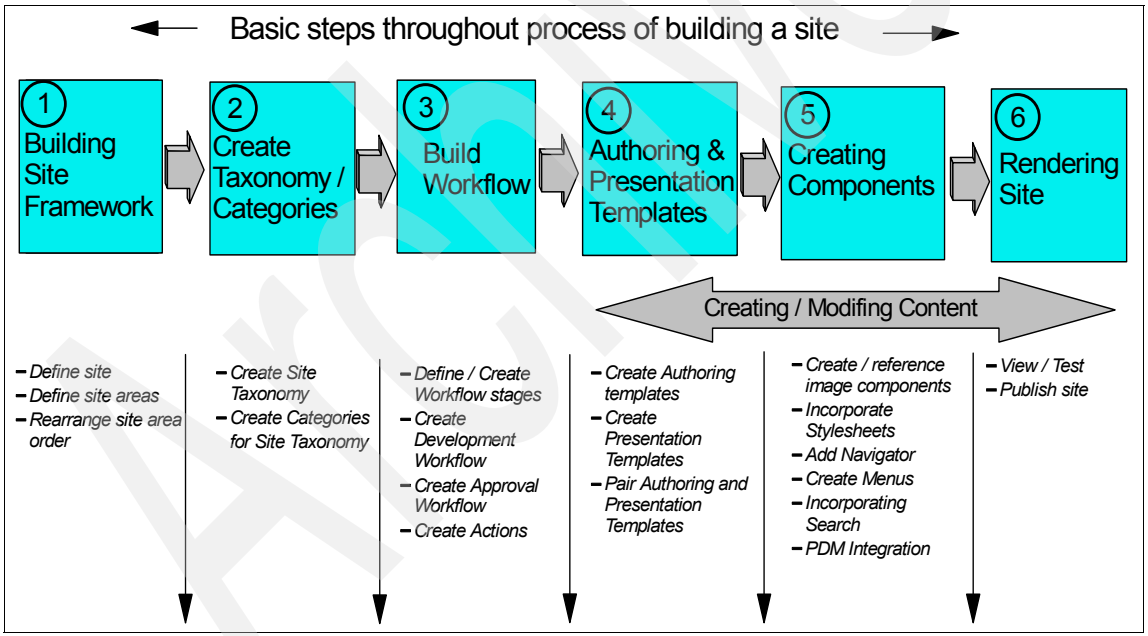


Figure 7-2 Overview of process for building a site

Archived

7.3 Prerequisites for building the example Web site

The following prerequisites are required:

- IBM Workplace Web Content Management 2.5 or 5.1 installed and configured as documented in Chapter 3, “Installing and configuring IBM Workplace Web Content Management” on page 69 or Chapter 4, “Overview of IBM Workplace Web Content Management 2.5” on page 125.
- Security enabled is not necessary.
- The Your Workflow Approver Group (the Book uses the Group named: *WCM Approval Managers*) created in WebSphere Member Manager or LDAP.
- Use of DB2 as the content repository is not necessary for a demo and proof of the concept scenario. However, it is recommended for production. For a guide on how to set up DB2 as a repository for IBM Workplace Web Content Management, see 3.3, “Configuring IBM Workplace Web Content Management with DB2 Universal Database” on page 77.
- Sample images, files, and code that used in the exercises, which you can download according to directions listed in Appendix A, “Additional material” on page 617.

Tip: The resources that we use to build the sample site, including images, code samples, and stylesheets, are available for download from the following:

`ftp://www.redbooks.ibm.com/redbooks/SG246792/`

7.4 Accessing IBM Workplace Web Content Management

The IBM Workplace Web Content Management user interface is accessible through a a Web browser, using the proper URL to the IBM Workplace server. To access IBM Workplace Web Content Management:

1. Open an Internet Web browser application, such as Microsoft Internet Explorer.

Note: The minimum browser requirement is Microsoft Internet Explorer version 5.5 with Service Pack 2 or Internet Explorer 6.0 with Service Pack 1.

Archived

2. In the Address line, enter the following URL:

   `http://<Hostname>:<Port>/wps/portal`

   In this URL:

   – *<Hostname>* is the fully qualified host name of your WebSphere Portal Server.

   – *<Port>* usually is 9081. For example, in our testing environment, the URL to access the server was as follows:

     `http://wcm02.cam.itso.ibm.com:9081/wps/portal`

3. Enter your user ID and password in the appropriate boxes.
4. Click **Login**.
5. Select the Web Content Management tab.

> **Note:** If for some reason, you do not see the Web Content Management tab when you log into WebSphere Portal, refer to Chapter 3, “Installing and configuring IBM Workplace Web Content Management” on page 69 to verify that IBM Workplace Web Content Management 5.1. is installed and access to the IBM Workplace Web Content Management authoring portlet is set correctly.

When you click this tab, you see the Web Content Authoring screen, similar to that shown in Figure 7-3.

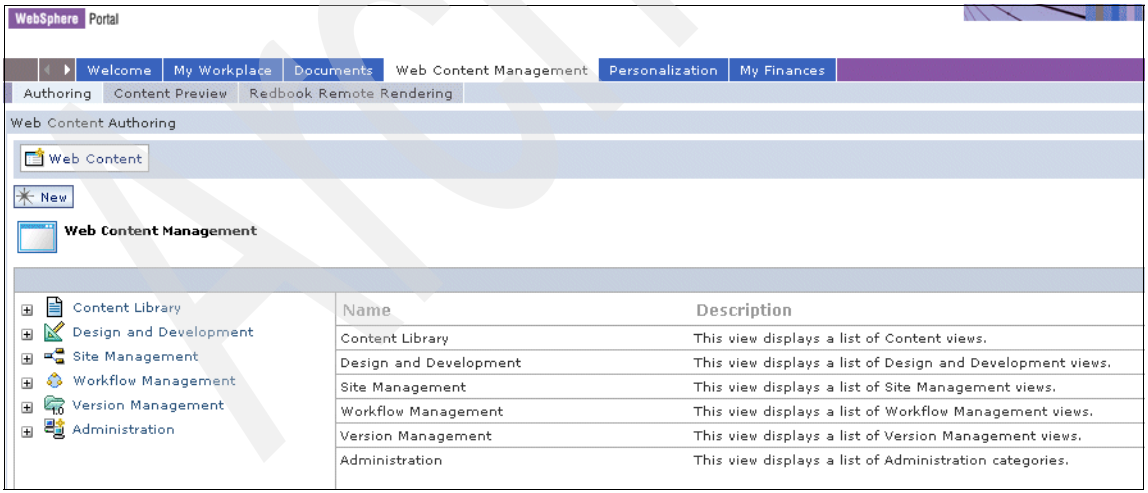


Figure 7-3 The Web Content Authoring screen

Note: On the left side of this screen, you see the items view navigator (Figure 7-4). You use this navigator to get to many of the components with which you work while building the site.

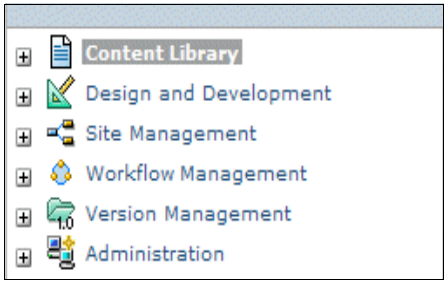


Figure 7-4 Items view navigator

7.5 Building the site framework

This section describes how to build the site framework. The following steps comprise the tasks that are required to complete this exercise.

- Creating the site
- Creating the site areas
- Rearranging the site area order

7.5.1 Creating the site

In this guided practice, you build a predefined site framework that is based on the River Bend Coffee and Tea Company example. To create the River Bend Coffee and Tea Company site, follow these steps:

1. In the item views navigator, expand Site Management.
2. Click **New**.
3. Select **Site**.
4. Click **OK**.
5. In the Name field, type `River Bend`, as shown in Figure 7-5 on page 260.

Archived

Figure 7-5 Naming the site

6. Click **Save…**, and then click **Save and Close**.
7. At this point, you see the River Bend site displayed in the index list. (Figure 7-6).

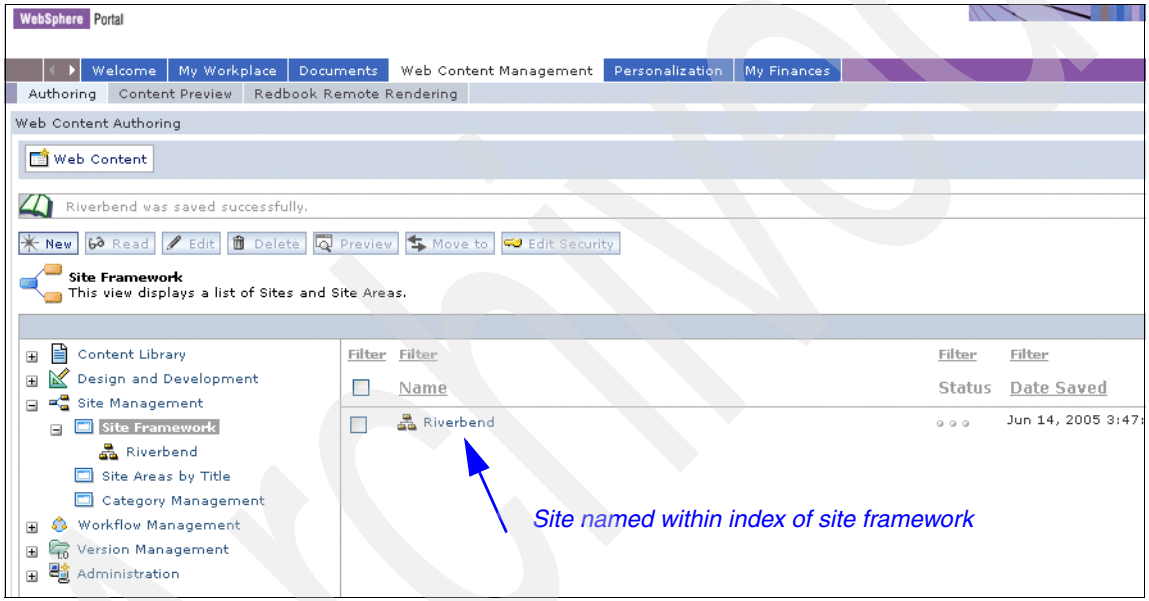


Figure 7-6 The River Bend site in the index of site framework

7.5.2 Creating the site areas

After you have created the site, you create site areas for each area within the site by following these steps:

1. Expand the site framework in the item views navigator, if it is not already expanded.
2. Click **New**.
3. Select **Site Area**.
4. Click **OK**.

Archived

5. Select **River Bend** as the parent site.

   Notice that the Start for the Link Order of the new site area is selected by default. Keep this selection.

> **Note:** As you build your site framework, make sure that you select the correct parent site with which to associate the different site areas. Select the correct site area, as shown in Figure 7-6 on page 260. The parent list contains all previously defined sites and site areas. When you select a site or site area from this list, you define under which parent your new site area is located.

6. Click **OK**.
7. In the Name field, enter `Home`.
8. Click **Save…,** and then click **Save and Close**.

   At this point, if you expand the site framework view, you see the home site area within the navigational hierarchy, as shown in Figure 7-7.

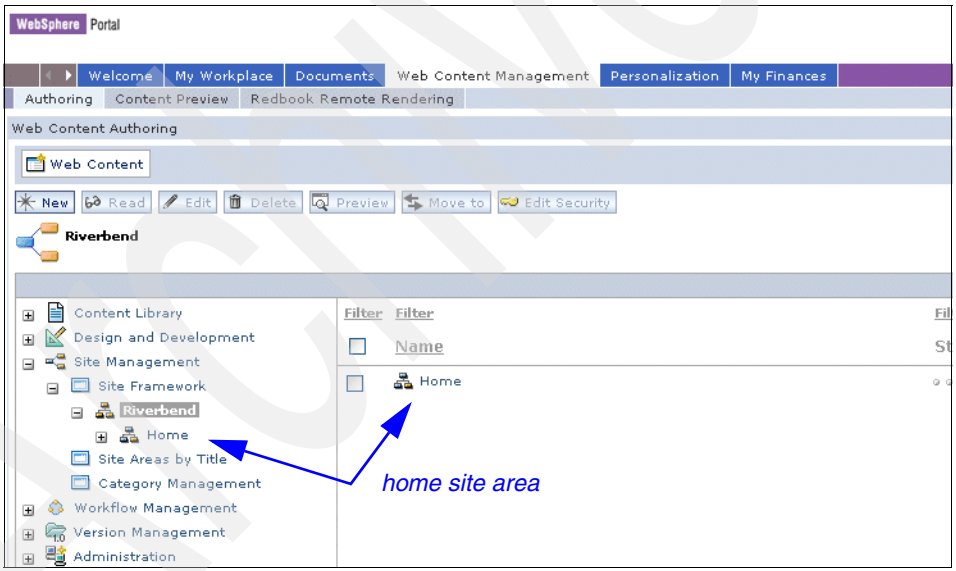


Figure 7-7 Creating the home site area

9. Click **New**.
10. Select **Site Area**, and click **OK**.
11. Expand **Riverbend** to show its dependant site areas.
12. Select **Home** as the parent site.

13. Select **End** for the link order of the new site area.

> **Note:** Selecting either Start or End for the link order determines the sequential placement of a site area within the list of multiple site areas. If the order of the site areas listed does not match the exact example that we provide, do not worry. Refer to 7.5.3, “Rearranging the site area order” on page 265 for details about how to re-order site areas.

14. Click **OK**.

15. In the Name field, enter the appropriate name of the site area. For example, for the first site area under Home, enter `Tools`, as shown in Figure 7-8.

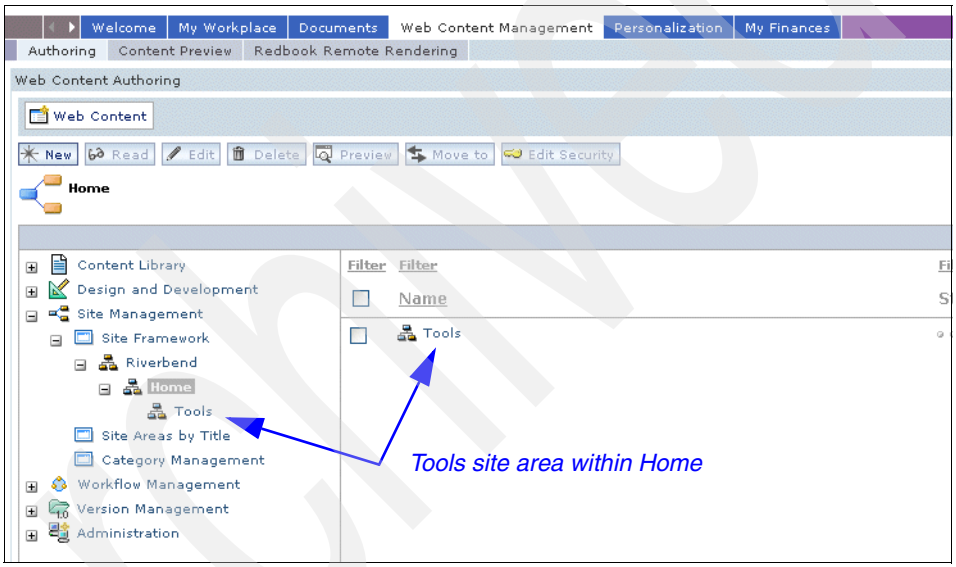


Figure 7-8 Creating the Tools site area within Home

16. Click **Save…**, and then click **Save and Close**.

At this point, you can repeat this process to create the site areas that are shown in Figure 7-9 on page 263. Repeat step 9 on page 261 through step 16 to create all the site areas for the site example.

> **Tip:** Use **Save ...** or **Save As** to create copies of an open object (that is, site area).

Attention: Make sure that you select the correct parent under which the site areas are located. In order to arrive at the same hierarchy for the site framework as that shown in Figure 7-9, you need to be sure to select the proper parent site when you create new site areas.

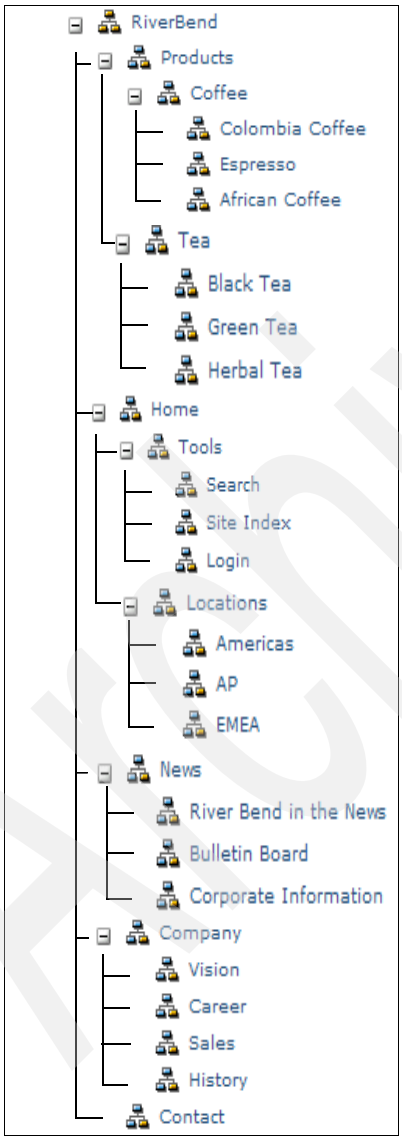


Figure 7-9 The RiverBend site framework

Archived

Tip: When you create additional site areas, we recommend that you start with the highest level nodes of the hierarchy. For example, create the site areas for Products, News, Company, and Contact as nodes beneath the parent River Bend. When you have created these, then proceed to make lower-level site areas beneath each of these parents. For example, you can make a site area for *Coffee* and a site area for *Tea* using *Products* as the parent. Finally, create the next sub-level site areas beneath each of the Coffee or Tea product categories.

You do not need to create all the site areas to complete the exercise. For the purpose of creating the River Bend Web site, the site areas marked with an asterisk (*) are mandatory, and the rest are optional:

River Bend*

- Products*
  - Coffee*
    - Columbia Coffee*
    - Espresso*
    - African Coffee*
  - Tea *
    - Black Tea
    - Green Tea
    - Herbal Tea
- Home*
  - Tools*
    - Search*
    - Site Index*
    - Login*
  - Locations
    - Americas
    - AP
    - EMEA
- News*
  - River Bend in the News*
  - Bulletin Board*
  - Corporate Information*
- Company
  - Vision
  - Career
  - Sales
  - History
- Contact

Archived

7.5.3 Rearranging the site area order

After you create a site area, you can change where the site area displays.

> **Note:** Within this section, we describe specific steps for rearranging the order of two sibling and children site area nodes within the site framework. Keep in mind that you can use the Move To button to move a specific site area to a different level or location within the hierarchy.

Follow these steps to move the Login site area to the beginning and back to the end position in the Tools site area on the River Bend site:

1. In the item views navigator, expand the following:
   - The Site Management view
   - The Site Framework view
   - The River Bend site
   - The Home site area
   - The Tools site area.
2. Select the **Login** site area.
3. In the button bar, click **Move To**.
4. Expand the River Bend site.
5. Expand the Home site area.
6. Expand the Tools site area.
7. Select **Login**.
8. Click **Move to**.
9. Expand the River Bend site, the Home site area, and the Tools site area.
10. Select **Search.**
11. Click **Previous** under Move Order.
12. Click **OK**.

The Login site area is now displayed before the Search site area within the list of site areas, as shown in Figure 7-10.

Figure 7-10 Rearranging the Login site area

Archived

13. To return the Login site area to its original position, select the **Login** site area.
14. Click **Move To**.
15. Expand the River Bend site framework.
16. Expand the Home site area.
17. Expand the Tools site area.
18. Select **Site Index**.
19. Click **Next** within the Move Order options.
20. Click **OK**.

The Login site area is now the last site area under the Tools site area.

7.6 Building the taxonomy and categories

This section explains how to build the taxonomy for the River Bend Web site. Refer to Chapter 5, “Key concepts and terminology” on page 141 for an overview of these components and a basic understanding of what you are going to build.

> **Note:** The process of building the taxonomy and categories which you perform in this section is a key step for creating a informal structure for your metadata. Consider this a behind-the-scenes requirement for organizing your content.

7.6.1 Creating the site taxonomy

Follow these steps to create the taxonomy for the River Bend Coffee and Tea Company Web site:

1. In the item views navigator under the Site Management view, click **Category Management**.
2. Click New.
3. Select **Taxonomy** from the New list.
4. Click **OK**.
5. In the Name field, enter `River Bend Web Site Categories`.
6. Click **Save…,** and then click **Save and Close.**

Archived

7.6.2 Creating categories for the site taxonomy

Follow these steps to create the categories:

1. In the item views navigator, click **Category Management** if it is not already selected.
2. Click **New**.
3. Select **Category** from the New list, if it does not already appear in the field.
4. Click **OK**.
5. Select **River Bend Web Site Categories** as the parent taxonomy.
6. Click **OK**.

> **Note:** For the first category, select River Bend Web Site Categories. For the rest, select the correct parent category, as shown in the following list.

7. In the Name field, enter `Corporate`.
8. Click **Save…,** and then click **Save and Close**.
9. Repeat Steps 2 through 8 to create all the categories for the Web site, using the following list as a guide.

The index list contains all previously defined taxonomies and categories. When you select a taxonomy or category from this list, you define under which parent your current category is located. The following lists the finished category taxonomy:

- Corporate
  - Employees
  - News
    - External
    - Internal
  - Policies
- Countries
  - Ceylon
  - China
  - Colombia
  - Kenya
- Products
  - Coffee
  - Espresso
  - Tea
- Stores
  - Catering*
  - Retail*

Archived

7.7 Building a workflow

The XYZ group in the River Bend Coffee and Tea Company needs the ability to view data. To review data each document has to go through a workflow after it is created. The workflows that you build in this exercise are used in the authoring template that you will build in the next section.

The following steps comprise the tasks that required to complete this exercise:

- Creating the intranet workflow stages
- Creating the publish stage
- Creating the Development Express Workflow
- Creating the Approval Workflow
- Creating actions

7.7.1 Creating the intranet workflow stages

Follow these steps to create the intranet workflow stages:

1. Click **Workflow Management** in the items view navigator.
2. Click **New**.
3. Select **Workflow Stage**.
4. Click **OK**.
5. In the Name field, type `Draft`.
6. Click **Save**.

   You see the following message above the button bar:

   ```
   Draft was saved successfully.
   ```

7. Click **Save...**, then click **Save and Close**.
8. Repeat Steps 1 through 7 to create the following stages:
   - Expired
   - Reject
   - Approval

Archived

Note: If you have created the *WCM Approval Managers* group in WebSphere Member Manager or LDAP continue with this exercise. To continue with the exercises in this chapter, it is not required to have this group. However, for an implementation or proof of concept, you need to implement a granular security framework that meets the needs of your company or customer.

If you only use wpsadmin for the rest of the chapter's exercises, you can skip to the next section, because wpsadmin has sufficient rights to perform all actions including approval.

9. Click **Grant Approve Access** in the Workflow Security section.
10. Click **Add**.
11. Select **Groups** from the User Type list.
12. Click Search.
13. Select **WCM Approval Managers** approval group.

Note: Only members of the IBM Workplace Web Content Management Approval Managers group are able to access that content that is in this stage.

14. Click **OK**.
15. Click **OK**.
16. Click **Save**, and then **Close**.

7.7.2 Creating the publish stage

Follow these steps to create a stage called *Development* that changes the status of content from *Draft* to *Published*:

1. Click **Workflow Management** in the items view navigator.
2. Click **New**.
3. Select **Workflow Stage** from the New list.
4. Click **OK**.
5. In the Name field, type `Publish`.
6. In the Properties section, click **Select Actions** under Execute on Entering Stage to select an action that is performed on any content that enters this stage of the workflow.

Archived

7. Click **Add**.

   The Execute on Entering Stage dialog box opens. It lists all the available actions. By default, the list includes Publish, Expire, and Publish and Expire

8. Select the **Publish** action.
9. Click **OK**.
10. Click **OK**.

> **Note:** You see the word *Publish* under Execute on Entering Stage.

11. Click **Save...**, and then click **Save and Close**.

7.7.3 Creating the Development Express Workflow

The Development Express Workflow is the express workflow that publishes content immediately without approval. We use this workflow for this exercise to avoid time consuming approval activities. Make sure that you are in the Workflow Management view in the Item Views Navigator. Then, follow these steps to create the Development Workflow:

1. Click **New**.
2. Select **Workflow** from the New list.
3. Click **OK**.
4. In the Name field, type `Development Express`.
5. Click Select **Workflow Stages** in the Properties section.
6. Click **Add**.
7. Select **Draft** and **Publish** from the list of workflow stages.
8. Click **OK**.
9. Click **OK**.
10. Click **Save...**, and then click **Save and Close**.

Archived

7.7.4 Creating the Approval Workflow

This guided practice creates a workflow that publishes content with approval, expiry, and reject stages. Make sure you are in the Workflow Management view in the Item Views Navigator. Then, follow these steps to create the Approval Workflow:

1. Click **New**.
2. Select **Workflow** from the New list.
3. Click **OK**.
4. In the Name field, type `Approval Workflow`.
5. Click **Select Workflow Stages** in the Properties section.
6. Click **Add**.
7. Select the following required stages an click **OK**:
   - Approval
   - Draft
   - Expired
   - Publish
8. Select the **Draft** stage.
9. Use the Move arrows, located to the right of the Workflow Stages list, to move the Draft stage to the top of the list.
10. Repeat steps 8 and 9 to rearrange the Workflow stages as follows:
    - Draft
    - Approval
    - Publish
    - Expired
11. Click **OK**.
12. Click **Select Reject Stage**.
13. Select the **Reject** stage.
14. Click **OK**.
15. Click **Save...**, and then click **Save and Close**.

Archived

7.7.5 Creating actions

Now that you have created the workflow, you need to create actions for the stages that move documents through the approval workflow. To create actions, you need to do the following:

- Assign the Expire action to the Expired stage.
- Create a scheduled move action.
- Assign the Scheduled Move action to the Publish stage.
- Create the e-mail action.
- Assign the e-mail action to the Expired stage.

Assigning the Expire action to the Expired stage

Make sure you are in the Workflow Management view in the items view navigator. Then, follow these steps to assign the Expire action to the Expired stage:

1. Click **Workflow Stages**.
2. Select the **Expired** stage.
3. Click **Edit**.
4. Click **Select Actions** under Execute on Entering Stage in the Properties section.
5. Click **Add**.
6. Select **Expire** from the Available list.
7. Click **OK**.
8. Click **OK**.
9. Click **Save and Close**.

Creating a Scheduled Move Action

Make sure you are in the Workflow Management view in the items view navigator. Then, follow these steps to create a Scheduled Move Action:

1. Click **New**.
2. Select **Scheduled Move Action**.
3. Click **OK**.
4. In the Name field, type `Move on Expiry Date`.
5. Select **Expiry Date** from the Date Type list.
6. Choose an appropriate date to see this action, for example tomorrow's date.
7. Click **Save and Close**.

   You see the Move on Expiry Date action in the Workflow Actions view.

Archived

Assigning the scheduled Move action and Publish action to the Publish stage

Make sure you are in the Workflow Management view in the items view navigator. Then, follow these steps to assign the Move on Expiry Date action that you created and the Publish action to the Published stage:

1. Click **Workflow Stages**.
2. Select the **Publish** stage.
3. Click **Edit**.
4. Click **Select Actions** under Execute on Entering Stage in the Properties section.
5. Click **Add**.
6. Select **Publish** and **Move on Expiry Date** from the list of available actions.
7. Click **OK**.
8. Click **OK**.
9. Click **Save and Close**.

Creating the e-mail action

Make sure you are in the Workflow Management view in the items view navigator. Then, follow these steps to create the e-mail action:

1. Click **New**.
2. Select **E-mail Action**.
3. Click **OK**.
4. In the Name field, enter `E-mail for Rejected Content`.
5. **Select E-mail Authors** in the E-mail Action section.

Note: E-mail Stage Approvers is selected automatically.

Type the following text in the Additional E-mail Text field:

```
Your content has been rejected. Please contact the content approver for
further details.
```

Note: This is the text the Content Author sees.

6. Click **Save** and **Close**.

Archived

Assign the e-mail action to the Reject stage

Make sure you are in the Workflow Management view in the items view navigator. Then, follow these steps to assign the e-mail action that you created in to the Expired stage:

1. Click **Workflow Stages**.
2. Select the **Reject** stage.
3. Click **Edit**.
4. Click the **Select Actions** under Execute on Entering Stage.
5. Click **Add**.
6. Select **E-mail for Rejected Content** from the list of available actions.
7. Click **OK**.
8. Click **OK**.
9. Click **Save and Close**.

7.8 Creating an authoring template

To create an authoring template, you need to:

- Create the authoring template named *River Bend Development*.
- Copy the River Bend Development authoring template to the River Bend External News Authoring Template.

7.8.1 Creating the River Bend Development authoring template

To create the River Bend Development authoring template, follow these steps:

1. Click **Design and Development** in the items view navigator.
2. Click **New**.
3. Click **Authoring Template.**
4. Click **OK**
5. In the Name field, type `River Bend Development`.
6. Click **Select Workflow** in the Workflow section.
7. Click **Development Express**.
8. Click **OK**.

Archived

Note: This action defines the default workflow for content that uses the River Bend Development authoring template, which is an express workflow without approval.

9. Click **Component Manager** at the top of the page.
10. Click **Add**.
11. In the Name field, enter `Display Title`.
12. Select **Text Component** from the Component Reference list.
13. Click **OK**.
14. Repeat Steps 10 through 13 to create the components that are listed in Table 7-1.

Table 7-1 Creating the component name and component type

Component name	Component type
Summary	Text Component
Body	Rich Text Component
Content Image	Image Component
PreBody	Component Reference
PostBody	Component Reference

The new component fields are added to the authoring template.

Note: Take care to avoid typographical errors, especially for blanks that separate two-word field names. Typographical errors are the main reason that content is not displayed as expected.

15. Click **OK**.
16. Click **Save...**, and then click **Save and Close**.

The River Bend Development authoring template appears in the index list.

Archived

7.8.2 Copying the River Bend Development authoring template

Follow these steps to create a copy of the River Bend Development authoring template:

1. Click **Authoring Templates** in the items view navigator.
2. Click **River Bend Development**.
3. Click **Edit**.
4. Click **Save...; Save As**.
5. In the Name field, type `River Bend External News`.
6. Click **OK**.
7. Expand the Workflow section.
8. Click **Select Workflow**.
9. Select **Approval Workflow.**
10. Click **OK**.
11. Click **Save…**, and then click **Save and Close**.

7.9 Creating the presentation templates

In this exercise, you create the following presentation templates:

- A three-column table presentation template to display most of the content files and a related links menu, such as Login, Tools, and Site Map.
- A two-column table presentation template to format content in site areas that do not require the related links menu, such as Login, Tools, and Site Map.
- A two-column table presentation template that contains the connect tag in order to obtain search results in the Search site area.

> **Note:** You can find the resources that we use to build the sample site, including images, code samples, and stylesheets at:
>
> `ftp://www.redbooks.ibm.com/redbooks/SG246792/`

After creating the presentation templates, you pair them appropriately.

Archived

7.9.1 Creating the presentation templates

Create the three-column table presentation template by following these steps:

1. In the items view navigator, expand **Design and Development**.
2. Click **New**.
3. Select **Presentation Template**.
4. Click **OK**.
5. In the Name field, enter `Three Column Table`.
6. In the Presentation Template section, replace the existing code with the code shown in Example 7-1.

Example 7-1 Three-column table presentation template code

```
<html>
<head>
<!--  insert River Bend stylesheet component reference here -->

<!-- insert search component reference here -->

</head>
<body text="black" bgcolor="white" background="" leftmargin="0" rightmargin="0"
topmargin="0" bottommargin="0">
<!-- page header table -->
<table border="0"  cellspacing="0" cellpadding="0"  width="760">
<tr>
<td  height="142">
<!-- insert River Bend logo and title image component references here -->

</td>
</tr></table>
<!--  main page table -->
<table class="middleBackground"  border="0"  width="760">
<tr>
<td colspan="3">
<!-- insert breadcrumb navigator component reference here  -->

<!-- insert long brown line image component reference here  -->

</td>
</tr>
<td valign="top" width="25%">
<!-- insert left navigator component reference here -->

<!-- insert tools navigator component reference here -->

<!-- insert quick search form html component reference here -->
```

Archived

```
</td>
<td  valign="top">
<!-- insert component tag references here -->

<!-- insert related category menu component reference here -->

</td>
<td  valign="top" align="center" width="20%" >
<!-- insert related information menu component reference here -->

</td>
</tr>
<tr>
<td colspan="3" >
<!-- insert long brown line image component reference here  -->

<!-- insert JSP component reference for displaying site messages here -->

</td>
</tr>
<tr>
<td colspan="3">
<!-- insert River Bend footer image component reference here -->

</td>
</tr>
</table>
</body>
</html>
```

7. Find the following comment in which to insert the component tag references which reference the Authoring Template's field values:

```
<!-- insert component tag references here -->
```

8. Place the cursor beneath the comment and enter the code shown in Example 7-2.

Example 7-2 Three-column table presentation template component tag reference code

```
<font class="contrastHeading">
<AptrixCmpnt context="current" key="Display Title" type="content"/><br>
</font>
<AptrixCmpnt context="current" key="Summary" type="content"/>
<AptrixCmpnt context="current" key="PreBody" type="content"/>
<AptrixCmpnt context="current" key="Body" type="content"/>
<AptrixCmpnt context="current" key="Content Image" type="content"/>
<AptrixCmpnt context="current" key="PostBody" type="content"/>
```

Archived

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

7.9.2 Creating the two-column table presentation template

Follow these steps to create the two-column table presentation template:

1. In the item views navigator, expand **Design and Development**.
2. Click **New**.
3. Select **Presentation Template** in the New drop-down list.
4. Click **OK**.
5. In the Name field, enter `Two Column Table`.
6. In the Presentation Template section, replace the existing code with the code in Example 7-3.

Example 7-3 Two-column table presentation template code

```
<html>
<head>
<!-- insert River Bend stylesheet component reference here -->

</head>
<body text="black" bgcolor="white" background="" leftmargin="0" rightmargin="0"
topmargin="0" bottommargin="0">
<!-- page header table -->
<table border="0"  cellspacing="0" cellpadding="0"  width="760">
<tr>
<td  height="142">
<!-- insert River Bend logo and title image component references here -->

</td>
</tr></table>
<!--  main page table -->
<table class="middleBackground"  border="0"  width="760">
<tr>
<td colspan="2">
<!-- insert breadcrumb navigator component reference here  -->

<!-- insert long brown line image component reference here -->

</td>
</tr>
<td valign="top" width="25%">
<!-- insert left navigator component reference here -->

<!-- insert tools navigator component reference here -->
```

Archived

```
<!-- insert quick search form html component reference here -->

</td>
<td  valign="top">
<!-- insert component tag references here -->

</td>
</tr>
<tr>
<td colspan="2" >
<!-- insert long brown line image component reference here -->

<!-- insert JSP component reference for displaying site messages here -->

</td>
</tr>
<tr>
<td colspan="2">
<!-- insert River Bend footer image component reference here -->

</td>
</tr>
</table>
</body>
</html>
```

7. Insert the component tag references. Find the following comment:

```
<!-- insert component tag references here -->
```

8. Place the cursor beneath the comment and enter the code shown in Example 7-4.

Example 7-4 Two-column table presentation template component reference code

```
<font class="contrastHeading">
<AptrixCmpnt context="current" key="Display Title" type="content"/><br>
</font>
<AptrixCmpnt context="current" key="Summary" type="content"/>
<AptrixCmpnt context="current" key="PreBody" type="content"/>
<AptrixCmpnt context="current" key="Body" type="content"/>
<AptrixCmpnt context="current" key="PostBody" type="content"/>
```

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

Archived

7.9.3 Creating the two-column table search presentation template

Follow these steps to create the two-column table search presentation template:

1. In the item views navigator, expand **Design and Development**.
2. Click **New**.
3. Select **Presentation Template** in the New drop-down list.
4. Click **OK**.
5. In the Name field, enter `Two Column Table Search`.
6. In the Presentation Template section, replace the existing code with the code shown in Example 7-5.

Example 7-5 Two-column table search presentation template

```
<html>
<head>
<!-- insert River Bend stylesheet component reference here -->

</head>
<body text="black" bgcolor="white" background="" leftmargin="0" rightmargin="0"
topmargin="0" bottommargin="0">
<!-- page header table -->
<table border="0"  cellspacing="0" cellpadding="0"  width="760">
<tr>
<td  height="142">
<!-- insert River Bend logo and title image component references here -->

</td>
</tr>
</table>
<!--  main page table -->
<table class="middleBackground"  border="0"  width="760">
<tr>
<td colspan="2">
<!-- insert breadcrumb navigator component reference here  -->

<!-- insert long brown line image component reference here -->

</td>
</tr>
<tr>
<td valign="top" width="25%">
<!-- insert left navigator component reference here -->

<!-- insert tools navigator component reference here -->

<!-- insert quick search form html component reference here -->
```

Archived

```
</td>
<td  valign="top">
<!-- insert component tag references here. Note that the Search Results title
and connect tag will be inserted above the PostBody component. -->

<!-- insert Search Results title here -->

<!-- format search results according to template specified in template
attribute of connect tag below -->

</td>
</tr>
<tr>
<td colspan="2" >
<!-- insert long brown line image component here -->

<!-- insert JSP component reference for displaying site messages here -->

</td>
</tr>
<tr>
<td colspan="2">
<!-- insert River Bend footer image component reference here -->

</td>
</tr>
</table>
</body>
</html>
```

7. Insert the component tag references. Find the following comment:

```
<!-- insert component tag references here -->
```

8. Place the cursor beneath the comment and enter the code shown in Example 7-6.

Example 7-6 Two-column table search presentation template component tag

```
<font class="contrastHeading">
<AptrixCmpnt context="current" key="Display Title" type="content"/><br>
</font>
<br>
<AptrixCmpnt context="current" key="Summary" type="content"/>
<br>
<br>
<AptrixCmpnt context="current" key="PreBody" type="content"/>
<AptrixCmpnt context="current" key="PostBody" type="content"/>
```

Archived

9. Select **Process Connect Tags**.
10. Click **Save…**, and then click **Save and Close**.

> **Note:** In the two-column table search presentation template, in the component tag reference code, there is no Body component reference. In this case, the most logical position for the PostBody component would be after the `<connect>` tag, because the search results are a type of a Body, at least from an appearance viewpoint. Also, the PostBody component is not actually used, but it is in here to provide symmetry with the PreBody component and to maintain consistency with the other presentation templates.

7.10 Pairing the authoring template and the presentation templates

Follow these steps to pair the authoring templates and presentation templates appropriately for the River Bend site and its various site areas:

1. In the items view navigator, expand **Site Management**.
2. Expand **Site Framework**.
3. Select **River Bend**.
4. Click **Edit**.
5. In the Properties section, click **Edit Template Mapping**.
6. Click **Add**.

   A list of the available authoring templates and presentation templates displays, as shown in Figure 7-11 on page 284.

Archived

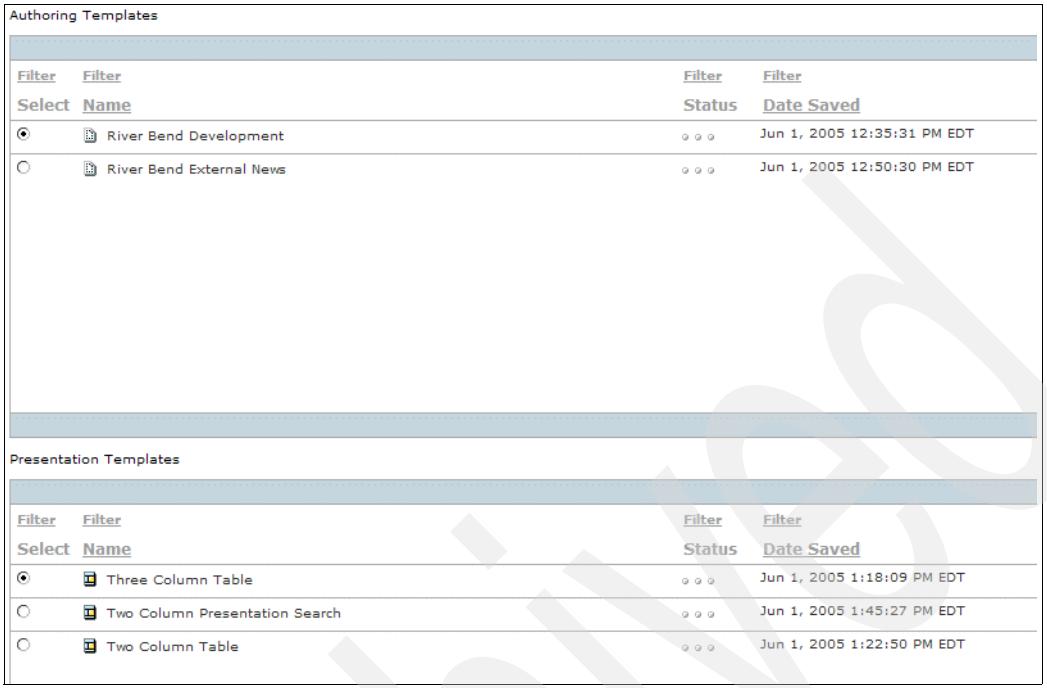


Figure 7-11 Pairing the authoring template and the presentation template

7. In the Authoring Templates section, select **River Bend Development**.
8. In the Presentation Templates section, select **Three Column Table**.
9. Click **OK**.

   The two templates are displayed as mapped.
10. Click **Add**.
11. Click **River Bend External News**.
12. Click **Three Column Table**.
13. Click **OK**.
14. Click **OK**.
15. Click **Save…**, and then click **Save and Close**.

Archived

16. Repeat Steps 3 through 15 to create the remainder of the template mappings for the site areas, according to Table 7-2. Be sure to select the appropriate site area for each template mapping. The site areas that are not mentioned in Table 7-2 inherit the template pairings.

Table 7-2 Site area pairing authoring template with presentation template

In this site area…	Pair this authoring template…	With this presentation template…
Tools	River Bend Development	Two Column Table
Site Index	River Bend Development	Two Column Table
Search	River Bend Development	Two Column Table Search
Login	River Bend Development	Two Column Table
River Bend in the News	River Bend External News	Three Column Table

7.11 Creating and referencing image components

To create and reference image components, you need to:

- Create several image library components.
- Reference these image components, as needed, in a presentation template and a content item. Note that one of these images is inserted when you create the left navigator later.

For this exercise, we used an image logo for the River Bend Web site.

Tip: The resources that we to build the sample site, including images, code samples, and stylesheets are available at:

`ftp://www.redbooks.ibm.com/redbooks/SG246792/`

For your implementation, you can use a stylesheet that reflects your company's corporate design.

7.11.1 Creating image library components

Follow these steps to add image library components.

1. In the items view navigator, expand **Design and Development**.
2. Click **Component Library**.
3. Click **New**.

4. Select **Image Component**, and then click **OK**.
5. In the Name field, enter `Riverbend_Logo`.

> **Note:** Use the same name as the image file so that you know what the library component contains.

6. In the Image Component section in the File section, click **Upload File**.
7. Click **Browse**.
8. **Browse** for the riverbend_logo.gif image in the location where you downloaded it.
9. Click **Open**, and then click **OK**.
10. Click **Save…**, and then click **Save and Close**.

    The image is saved in the library of image components.
11. Expand Image Component in the items view navigator to see the new image component that is listed in the index.
12. Repeat steps 3 through 11 to create the other image components as follows:
    - Riverbend_Title (uses riverbend_title.gif)
      - Set Width to 320
      - Set Height to 120
    - Riverbend_Footer (uses riverbend_footer.gif)
      - Set Width to 755
      - Set Height to 34
    - Short_Brown_180_Pixel_Line (uses brown_dot.gif)
      - Set Width to 180
      - Set Height to 1
    - Long_Brown_755_Pixel_Line (uses brown_dot.gif)
      - Set Width to 755
      - Set Height to 1
    - Coffeebeans (uses CoffeeSmall.jpg)
    - Tealeaves (uses TeaSilverNeedle.jpg)
    - Redbook_Team (uses RedbookTeam.jpg)
      - Set Width to 200
      - Set Height to 150

Archived

Note: For the line images, you use a single dot and then size it accordingly. The image brown_dot.gif is used for both of the different line sizes in this activity. When you specify values in the height and width fields for an image component that uses this type of image (a 1-pixel by 1-pixel dot), the browser scales the image according to those values. You can reuse the same image file to create lines of different sizes on your Web site. For the other images, you set the height and width during the save action, according to the images' values of the file.

7.11.2 Referencing images in a presentation template

Now, you are ready to insert the following image library components in a presentation template:

- Riverbend_Logo
- Riverbend_Title
- Riverbend_Footer
- Long_Brown_755_Pixel_Line

Follow these steps to insert the image library components as references in the Three Column Table presentation template:

1. In the items view navigator, expand **Design and Development**.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. Find the following tag in the Presentation Templates section:

   ```
   <!--insert River Bend logo and title image component references here-->
   ```

   Enter the following line of code beneath the tag:

   ```
   <AptrixLibCmpnt name="Riverbend_Logo"/>
   <AptrixLibCmpnt name="Riverbend_Title"/>
   ```

6. You eventually insert a breadcrumb navigator below the title and logo. Beneath where the breadcrumb navigator will be, you need to insert the long brown line image component reference. Find the following tag in the Presentation Templates section:

   ```
   <!-- insert breadcrumb navigator component reference here -->
   ```

   Beneath that tag, find this tag:

   ```
   <!-- insert long brown line image component reference here -->
   ```

   Enter the following line of code beneath the tag:

   ```
   <AptrixLibCmpnt name="Long_Brown_755_Pixel_Line"/>
   ```

Archived

7. Insert code for the second instance of the long brown line separator just above where the JSP component will be. In the Presentation Templates section, find the following comment:

   ```
   <!-- insert long brown line image component reference here -->
   ```

   Directly below this tag, enter the following code:

   ```
   <AptrixLibCmpnt name="Long_Brown_755_Pixel_Line"/>
   ```

8. Insert the River Bend footer. In the Presentation Templates section, find the following comment:

   ```
   <!-- insert River Bend footer image component reference here -->
   ```

   Enter the following line of code below this comment:

   ```
   <AptrixLibCmpnt name="Riverbend_Footer"/>
   ```

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

> **Note:** For the line images, you use a single dot and then size it accordingly. The image brown_dot.gif is used for both of the different line sizes in this activity. When you specify values in the height and width fields for an image component that uses this type of image (a 1-pixel by 1-pixel dot), the browser scales the image according to those values. You can reuse the same dot image file to create lines of different sizes on your Web site.

10. Repeat steps 1 through 9 to enter these image library component references into the Two Column Table and Two Column Table Search presentation templates, as appropriate for each template. These templates contain the same comments that are in the Three Column Table presentation template.

7.12 Creating content

To create content, you need to:

- Create a content item, add it to a site area, and associate a category with the content to test that these items work properly.
- Create a content item and assign it as the default content page for a site area.
- Create additional content items for the remaining site areas in the River Bend Web site.

Archived

7.12.1 Creating content to test that items in the site work properly

Follow these steps to create content to test your work:

1. In the item views navigator, expand Content Library.
2. Click **New**.
3. Select **Content**.
4. Click **OK**.

   The Web Content Authoring screen appears with a directive to choose your content's authoring template and site area. The River Bend Development authoring template displays in the Authoring Templates list.
5. In the Authoring Template field, select **River Bend Development**, if it is not already selected.
6. In the items view navigator in the Site Framework view, expand the River Bend site and the News site area.
7. Select **River Bend in the News**.
8. Click **OK**.

   An untitled object window opens.
9. In the Name field, enter `WCM Redbook Team in the News`. This is the name of the content file, not what is displayed in the Web browser.
10. In the Description field, enter `WCM Redbook Team in the News`.

Next, you need to associate a category with this content. Follow these steps:

1. Expand the **Profile** section.
2. Click **Select Categories**.
3. Click **Add**.
4. Expand the following:
   - River Bend Web Site Categories taxonomy
   - Corporate category
   - News category.
5. Select **Internal**.
6. Click **OK**.
7. Click **OK**.
8. In the Display Title field, enter `WCM Redbook Team in the News`.

   The Display Title is displayed in the News menu. The text in the Display Title is also displayed at the top of the content page.

Archived

9. In the Summary field, enter the following text:

```
News about the Team writing the Redbook in Cambridge 2005
```

Note: The summary text appears directly beneath the title on the content page. So, it does not need a period at the end of the sentence.

10. In the Body field, enter the following text:

```
The team who worked for four weeks in Cambridge, MA, to write the book
are (in the picture, top row, left to right) Michael Fromin, ascendant
Technologies, Denver CO and Amir Jordan, wpexperts, Cologne, Germany.
(Second row, left to right) Sabine Nagl, IBM Germany and Thomas
Radigewski, Germany. Not pictured: John Bergland, Project and Residency
Leader, IBM, Cambridge MA.

The amount of information that went into the book was substantial in
order to meet the expectations and requirements that have come up since
the last redbook was published in 2004. To achieve this target, a large
number of experts contributed to the book. Because these contributors
are located from Australia to Europe and North America, the book is a
worldwide achievement.
```

After you enter this text, it becomes the body of content for your page.

Note: If the text that you type into the Body field is one line in length or if it is multiple lines in the same paragraph, be sure to press Enter to ensure that the paragraph tags are in place. If not, the body text is appended to the summary text with no separation.

11. Use the Rich Text editor to format text. Select **Team** and make the text bold.
12. Click **Save**.

Note: After closing the content object a green symbol in the status column on the right side of the item views navigator shows that the content was immediately published. The Development Workflow defined in the authoring template pushes the content immediately to the Publish stage. It does not use an Approval stage.

Archived

7.12.2 Creating a content item and assigning it as the default content page for a site area

To create a content item, follow these steps:

1. Expand Content Library in the items view navigator.
2. Click **New**.
3. Select **Content**.
4. Click **OK**.

   The Web Content Authoring screen appears, with a directive to choose your content's Authoring Template and site area.
5. In the Authoring Template field, select **River Bend Development** if it is not already selected.
6. In the Site Framework view in the items view navigator, expand the River Bend site and News site area.
7. Select **River Bend in the News**.
8. Click **OK**.
9. In the Name field, enter River Bend in the News.
10. In the Description field, enter the following text:

    ```
    Default content page for River Bend in the News site area.
    ```
11. Expand the Profile section, and click **Select Categories**.
12. Click **Add**.
13. Expand the following:
    - River Bend Web Site Categories taxonomy
    - Corporate category
    - News category
14. Select **Internal** and **External**.
15. Click **OK**.
16. Click **OK**.
17. In the Display Title field, enter the following text:

    ```
    Recent news about River Bend
    ```
18. In the Summary field, enter the following text:

    ```
    Recent news about River Bend
    ```
19. In the Body field, enter the following text:

    ```
    Check this page often to see the latest news brewing at River Bend.
    ```

Archived

20. Click **Save…**, and then click **Save and Close**.

To assign this content item as the default content page for the River Bend in the News site area, follow these steps:

1. Expand Site Management in the items view navigator.
2. Expand Site Framework.
3. Expand the River Bend site and the News site area.
4. Select **River Bend in the News**.
5. Click **Edit**.
6. In the Properties section, under Default Content, click **Select Default Content**.
7. Select the River Bend in the News content page.
8. Click **OK**.
9. Click **Save**.
10. Click **Preview** to view the content.
11. Click **Save…**, and then click **Save and Close**.

Attention: At this point in the process, it is possible to view the Web site in the browser — without a portal and portlets — as a stand-alone Web site. You must add the default content to a site area before you can view the site area.

Open your browser, and use the following URL (for IBM Workplace Web Content Management 2.5 and 5.1):

```
http://<your servername>:<port>/wps/wcm/myconnect/RiverBend/News/
River+Bend+in+the+News
```

For example, we used the following URL for this book:

```
http://wcm01.cam.itso.ibm.com:9081/wps/wcm/myconnect/RiverBend/News/
River+Bend+in+the+News
```

7.12.3 Referencing an image component in a content item

Follow these steps to reference the image component of the content item named *WCM Redbook Team in the News*.

1. In the item views navigator, expand Content Library.
2. Click **Content by Title**.
3. Select the **WCM Redbook Team in the News** content item.
4. In the button bar, click **Edit**.

5. Scroll down to the PostBody section. Beneath Component, click **Select Component**.
6. In the items view navigator, click **Image Component**.
7. In the index, click **Redbook_Team**.
8. Click **OK**.
9. Click **Save**.
10. Click **Preview**.
11. Click **Refresh**.

Note: The image appears on the *WCM Redbook Team in the News* content page. The image is located in the PostBody section, beneath the paragraph that describes the members of the team.

Note: Regarding the image, Short_Brown_180_Pixel_Line, that you added to the component library but have not used yet, you will insert this image in the left navigator in the section where you create navigators later. The short brown line separates each of the site area links in the left navigator in the River Bend Web site that you are creating.

7.12.4 Creating content items for the remaining site areas in the River Bend Web site

The following tables contain information about the content for the River Bend Web site. You create content by using the information for the fields and by adding the content as default content to the site areas as indicated. For all content items, use the River Bend Development authoring template.

Important: You can only view a page or site area after assigning a default content.

Archived

Table 7-3 Default content for site area Products

Field	Default content
Title	Products
Category	Coffee, Tea
Display title	Products
Summary text	River Bend Products
Body text	Coffee and tea from selected plantations.
Other	Default content for Products site area

Table 7-4 Default content for site area Coffee

Field	Default content
Title	Coffee
Category	Coffee
Display title	Coffee
Summary text	River Bend Coffees
Body text	Our coffees are fair trade, organically grown, hand-roasted coffees. You can enjoy them at one of our retail outlets, or you can order them from our catalog and have them shipped to your home.
Other	Image component *Coffeebeans* in Postbody field; default content for Coffee site area

Archived

Table 7-5 Default content for site area Colombia Coffee

Field	Default content
Title	Colombia Coffee
Category	Colombia, Coffee
Display title	Colombia Coffee
Summary text	River Bend Colombia Coffees
Body text	Coffee beans from selected plantations in Colombia.
Other	Image component *Coffeebeans* in Postbody field; Default content for Colombia Coffee site area

Table 7-6 Default content for site area Columbia Coffee, Espresso

Field	Default content
Title	Colombia Coffee
Category	Colombia, Espresso
Display title	Rich Colombia Coffee from West Coast plantation
Summary text	Recent news about rich colombian espresso bean
Body text	Latest harvest created a rich, dark espresso bean due to excellent weather conditions.
Other	

Archived

Table 7-7 Default content for site area Espresso

Field	Default content
Title	Espresso
Category	
Display title	Coffee, Espresso
Summary text	River Bend Espresso
Body text	Our selection of strong and aromatic espresso beans from selected plantations.
Other	Image component *Coffebeans* in Postbody field; Default content for Espresso site area

Table 7-8 Default content for site area African Coffee

Field	Default content
Title	African Coffee
Category	Kenya
Display title	African Coffee
Summary text	River Bend African Coffees
Body text	Our selection of beans from plantations in traditional coffee growing areas in Africa.
Other	Default content for African Coffee site area

Archived

Table 7-9 Default content for site area Tea

Field	Default content
Title	Tea
Category	Tea
Display title	Tea
Summary text	River Bend's finest Tea selection
Body text	Our teas are fair trade and grown organically on the hills of China and Ceylon. Herbs for our Herbal Tea is grown organically in selected areas around the world.
Other	Image component *Tealeaves* in Postbody field; Default content for Tea site area

Table 7-10 Default content for site area Black Tea

Field	Default content
Title	Black Tea
Category	Tea, China, Ceylon
Display title	Black Tea
Summary text	River Bend's finest Black Tea selection
Body text	Our teas are fair trade and grown organically in the hills of China and Ceylon.
Other	Image component *Tealeaves* in Postbody field; Default content for Black Tea site area

Archived

Table 7-11 Default content for site area Green Tea

Field	Default content
Title	Green Tea
Category	Tea, China
Display title	Green Tea
Summary text	River Bend's finest Green Tea selection
Body text	Our teas are fair trade and are grown organically in the hills of China.
Other	Image component *Tealeaves* in Postbody field; Default content for Green Tea site area

Table 7-12 Default content for site are Herbal Tea

Field	Default content
Title	Herbal Tea
Category	Tea, Ceylon, China, Kenya
Display title	Herbal Tea
Summary text	River Bend's finest Herbal Tea selection
Body text	Herbs for our Herbal Tea are grown organically in selected areas around the world.
Other	Image component *Tealeaves* in Postbody field; Default content for Herbal Tea site area

Archived

Table 7-13 Default content for site area Home

Field	Default content
Title	Home
Category	Corporate
Display title	Welcome to River Bend Coffee and Tea Company
Summary text	A quick look at River Bend
Body text	Thank yo for visiting us today at River Bend Coffee and Tea Company.
Other	Default content page for Home site area

Table 7-14 Default content for site area Tools

Field	Default content
Title	Tools
Category	
Display title	Tools
Summary text	Site navigation and search tools
Body text	You can use the tools in this section to enhance your experience using the River Bend Web site.
Other	Default content page for *Tools* site area

Archived

Table 7-15 Default content for site area Search

Field	Default content
Title	Search
Category	
Display title	Search
Summary text	Search River Bend
Body text	Use this page to find what you are looking for on the River Bend Web site.
Other	Default content page for Search site area Verify that the PreBody section contains the HTML Search Form component

Table 7-16 Default content for site are Site Index

Field	Default content
Title	Search
Category	
Display title	Site Index
Summary text	River Bend's Web site at a glance
Body text	If you do not find what you are looking for, come back soon. We add new information about our coffees and teas daily.
Other	Default content page for Site Index site area; the PostBody component contains the NAV-Site Map component

Archived

Table 7-17 Default content for site area Login

Field	Default content
Title	Login
Category	
Display title	Login
Summary text	Access to River Bend IT systems
Body text	This page is for River Bend Coffee and Tea company employees and contractors.
Other	Default content page for Login site area; PostBody component contains the HTML Login Form component

Table 7-18 Default content for site area News

Field	Default content
Title	News
Category	News
Display title	News
Summary text	News of interest to River Bend employees, customers, and investors
Body text	Read about the latest news, product developments, and announcements at River Bend. This information changes often, so check back frequently.
Other	Default content page for News site area

Archived

Table 7-19 Default content for site area Bulletin Board

Field	Default content
Title	Bulletin Board
Category	Employees, External, Internal
Display title	Bulletin Board
Summary text	Important notices and announcements from River Bend
Body text	Visit the River Bend Bulletin Board to see interesting news for both employees and customers.
Other	Default content page for Bulletin Board site area

Table 7-20 Default content for site area Corporate Information

Field	Default content
Title	Corporate Information
Category	News
Display title	Corporate Information
Summary text	Facts about the River Bend business
Body text	River Bend considers itself a responsible member of the business community. As such, we donate 10% of our pre-tax profits to charitable causes. In addition, all of our employees are co-owners of River Bend Coffee and Tea Company.
Other	Default content page for site area Corporate Information

Important: Use the Home content item as default content for all remaining site areas.

Note: Create the content pages and assign them to the site areas as default content.

Archived

7.12.5 Opening and viewing the River Bend Web site in a browser

After you have assigned default content to the site areas, the River Bend Web site is ready for browsing.

> **Note:** You will add further design elements in the following sections to complete the design.

To open and view the River Bend Web site in a browser, follow these steps:

1. Open a Web Browser window.
2. Enter the following URL. You should replace <*your Servername*> with your actual values.

   ```
   http://<your Servername>:<port usually 9081>/wps/wcm/myconnect/RiverBend
   ```

   For example, the site for the example River Bend Web site is:

   ```
   http://wcm01.cam.itso.ibm.com:9081/wps/wcm/myconnect/RiverBend
   ```

> **Tip:** If you cannot view a certain page as you navigate the site, the default content for that page might be missing. Be sure that you have defined default content for all site areas.

7.13 Using stylesheets

To use stylesheets, you need to do the following:

- Save the stylesheet as a file resource component
- Reference the stylesheet component in the presentation templates

That stylesheet that we use for the River Bend Web site is available for download (riverbend.css). For information about how to download the stylesheet, see Appendix A, “Additional material” on page 617. For your own implementation, you will use a stylesheet that reflects your company’s corporate design.

> **Note:** The sample code that we use throughout the exercises within this chapter are available for download as described in Appendix A, “Additional material” on page 617.

Archived

7.13.1 Saving the stylesheet as a file resource component

Follow these steps to save the riverbend.css stylesheet as a file resource component:

1. In the item views navigator, expand Design and Development.
2. Expand the Component Library section.
3. Click **New**.
4. Select **File Resource Component**.
5. Click **OK**.
6. In the Name field, enter `riverbend.css`.
7. In the File Resource Component section, click **Upload File**.
8. Click **Browse**.
9. Select the riverbend.css stylesheet.
10. Click **Open**.
11. Click **OK**.
12. Click **Save…** and then click **Save and Close** to close the file resource component.

7.13.2 Referencing the stylesheet in the presentation templates

Follow these steps to reference the stylesheet in the presentation templates:

1. In the item views navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select the **Three Column Table presentation** template.
4. Click **Edit**.
5. In the Presentation Template section, find the following comment:

   ```
   <!-- insert River Bend stylesheet component reference here -->
   ```

   Insert the following code beneath the comment:

   ```
   <link rel="stylesheet" href='<AptrixLibCmpnt name="Riverbend.css"/>'>
   ```

   Attention: Carefully note the use of single (‘) and double (“) quotation marks.

6. Click **Save…** and then click **Save and Close** to save the Three Column Table presentation template.

Archived

> **Note:** Continue with the remainder of the steps only if you use IBM Workplace Web Content Management 2.5. If you use If you use IBM Workplace Web Content Management 5.1, simply refresh your browser to text the template.

7. Preview a content file that uses the Three Column Table presentation template to test your work, by expanding Content Library.
8. Click **Content by Title**.
9. Select the **Bulletin Board** content item.
10. Click **Preview**.

    When the content page opens, you can see the stylesheet has been applied because the background is light brown and the font in the summary and body text has changed.

    Follow steps 2 through 6 to add the stylesheet reference to the Two Column Table and Two Column Table Search presentation templates.

7.14 Adding a left navigator component

This section explains how to add a navigator component item that displays the site areas under the site River Bend on the left-hand side of the Web site. The navigator uses alternating design for selected and non-selected elements by applying HTML layouts. To display the navigator, the component reference is added to the three presentation templates.

To add a left navigator component, you need to do the following:

- Create the left navigator component.
- Reference the left navigator component in the presentation template.
- Format the left navigator component by creating bold and normal HTML components.
- Reference the HTML formatting components in the left navigator component.

7.14.1 Creating a left navigator component

To create the navigator that appears on the left side of every Web page, do the following:

1. Expand Design and Development in the items view navigator.
2. Expand Component Library.

Archived

3. Click **New**.
4. Select **Navigator Component**.
5. Click **OK**.
6. In the Name field, enter `NAV-Left`.

> **Tip:** It is good style and helpful to use a naming schema for components. It makes references easier to understand. For navigators, use a `NAV-` prefix. A reference then reads:
>
> ```
> <AptrixLibCmpnt name="NAV-Left"/>
> ```

7. In the Description field, enter `Left navigator for accessing site area sections`.

   > **Note:** The Navigator Component section contains fields for defining the navigator's:
   >
   > - Search criteria (in the top half of the window)
   > - Formatting (in the bottom half of the window)

8. In the Navigator Component section in the Start Type list, choose **Selected**.

   This action indicates that you define the site area from which the search begins.

9. In the Selected Start Area section, click **Select Start Area**.
10. In the index, click **River Bend**.
11. Click **OK**.
12. Deselect **Include Start** so that River Bend does not appear in the navigator.
13. From the Descendant Level list, select **1 Level**.

    This action defines that the first level of children site areas of River Bend are collected.

14. Deselect **Show Content**.

    > **Note:** If this option is selected, links to content in the current site area display in the navigator.

15. Select **Expand Current**.

Archived

Note: If this option is selected, the navigator expands to display the children site areas of the current site area.

16. In the Header field, enter the following code:

```
<font class="contrastHeading">
Site Navigation
</font>
```

17. Add short brown lines to separate the site area links. In the Separator field, enter the following line of code:

```
<AptrixLibCmpnt name="Short_Brown_180_Pixel_Line"/><br>
```

18. Indent each site area level and create a placeholder for the code you will add to format the site area links. In the Component Design 1 field, enter the following code:

```
<IndentCmpnt offset="0" repeat="   "/>
<placeholder tag="namelink"/><br>
```

19. Click **Save…** and then click **Save and Close**.

7.14.2 Referencing the left navigator in the presentation templates

Follow these steps to insert a tag to reference the left navigator component in the presentation templates:

1. In the item views navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. In the Presentation Template section, add the following comment:

```
<!--insert left navigator component reference here-->
```

   Insert the following code below the comment:

```
<AptrixLibCmpnt name="NAV-Left"/>
```

6. Save the presentation template.
7. Refresh the browser to view the left navigator on the Bulletin Board content page.
8. Click each navigator link to view the default content for each site area.

Archived

Note: You created the default content items and assigned a default content page to each site area in 7.12.4, “Creating content items for the remaining site areas in the River Bend Web site” on page 293.

9. Repeat steps 2 through 6 to reference the left navigator component in the Two Column Table and Two Column Table Search presentation templates.
10. Preview your work by clicking the Site Index (uses the Two Column Table presentation template) site area link and the Search (uses the Two Column Table Search presentation template) site area link in the left navigator.

7.15 Formatting the navigator

To format the navigator so that the current site area is formatted differently from the rest of the site areas in the navigator, you need to:

- Create HTML components to specify bold and plain formatting.
- Reference the bold and plain HTML formatting components in the left navigator component.

7.15.1 Creating HTML components to specify formatting

Follow these steps to create the bold and plain formatting components:

1. In the item views navigator, expand Design and Development.
2. Expand Component Library.
3. Click **New**.
4. Select **HTML Component**.
5. Click **OK**.
6. In the Name field, enter `HTML-Layout-Main Nav Bold`.
7. In the Description field, enter:

   ```
   This is the bold text formatting component.
   ```

8. In the HTML Component section, enter the following line of code:

   ```
   <a href='<placeholder tag="href"/>' >
   <b><placeholder tag="name"/></b>
   </a><br>
   ```

Attention: Be sure to use single (‘) and double (“) quotation marks correctly.

9. Click **Save** and then click **Save and Close**.
10. Repeat steps 3 through 9 to create a new HTML component with the following changes:
    - In the Name field, enter `HTML-Layout-Main Nav Plain`.
    - In the Description field, enter the following:

      ```
      This is the plain text formatting component.
      ```

    - In the HTML Component section, enter the following line of code:

      ```
      <placeholder tag="namelink"/><br>
      ```

7.15.2 Referencing the HTML formatting components in the left navigator component

To specify the format for bold and plain and to indent each level in the site navigation, do the following:

1. In the item views navigator, expand Component Library.
2. Expand Navigator Component.
3. Select **NAV-Left**.
4. Click **Edit**.
5. In the Navigator Component section, scroll to the Component Design 1 field.
6. Delete the existing placeholder tag in the field, and enter the following line of code:

   ```
   <IndentCmpnt offset="0"
   repeat="   "/>
   <AlternateDesign highlight="HTML-Layout-Main Nav Bold"
   normal="HTML-Layout-Main Nav Plain" type="Any"/>
   ```

7. Click **Save** and then click **Save and Close**.
8. Refresh your browser.
9. Navigate to River Bend in the News site area. Note how the current site area is formatted with bold text.

Archived

7.16 Creating a breadcrumb navigator

To create a breadcrumb navigator, you need to:

- Create a breadcrumb navigator component.
- Reference the breadcrumb navigator component in the presentation templates.

7.16.1 Creating the breadcrumb navigator component

To create the breadcrumb navigator that appears at the top of every Web page, just beneath the River Bend logo and title, do the following:

1. In the item views navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **Navigator Component**.
5. Click **OK**.
6. In the Name field, enter `NAV-Breadcrumbs`.
7. In the Description field, enter:

   ```
   This navigator appears at the top of every Web page.
   ```

8. In the Navigator Component section, select **Current - Site Area** from the Start Type list.

   This action indicates that the navigator needs to determine the site area of the currently displayed content.

9. Be sure that Include Start is selected.

   This action indicates that the navigator must include the first item it finds (for example, the site area of the current content).

10. Select **All** from the Ancestor Level list.

    Here, you define that you want to collect all the parent site areas of the current site area, all the way to the highest level.

11. Deselect Show Content.

Note: If this check box is selected, links to content in the current site area display in the navigator.

Archived

12. In the Component Design 1 field, enter the following code:

```
<a href='<placeholder tag="href"/>'><placeholder
tag="name"/></b></a>
```

This field specifies the parent site area.

13. Click **Add Component Design** at the bottom.

A new field is created, labeled Component Design 2.

14. In the Component Design 2 field, enter the following code:

```
 <b>&gt;</b> <a href='<placeholder
tag="href"/>'><b><placeholder tag="name"/></b></a>
```

This field specifies the design of the current site area. In the breadcrumb navigator, a greater than symbol (>) precedes the current site area.

15. Click **Save…** and then click **Save and Close**.

7.16.2 Referencing the breadcrumb navigator component in the presentation templates

To reference the breadcrumb navigator component in the presentation templates, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. In the Presentation Template section, look for the following comment in the code:

```
<!-- insert breadcrumb navigator component reference here -->
```

Enter the following line of code after the comment:

```
<AptrixLibCmpnt name="NAV-Breadcrumbs"/>
```

6. Click **Save…** and then click **Save and Close**.
7. To test your work, expand Content Library, select the River Bend in the News content item.
8. To view the result of your changes, refresh your browser.

On the Web page, the breadcrumb navigator is visible beneath the River Bend logo and title.

9. Repeat steps 3 through 6 to reference the bread crumb navigator in the Two Column Table and Two Column Table Search presentation templates.

Archived

Note: At this point, if you are using IBM Workplace Web Content Management 5.1, you can see the result of your work. Use the following URL to see River Bend as stand-alone Web site without portal framework:

```
http:/<hostname><port>/wps/wcm/myconnect/RiverBend
```

Port usually is 9081.

10. Test the breadcrumb navigator in the Two Column Table presentation template by previewing the Site Index content item in the browser.
11. Test the breadcrumb navigator in the Two Column Table Search presentation template by previewing the Search content item in the browser.

7.17 Creating a site map navigator

To create a site map navigator, you need to do the following:

- Create the site map navigator.
- Display the site map navigator on the Site Index content page.

7.17.1 Creating the site map navigator component

To create a site map navigator component that appears beneath the body text on the Site Index content page, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **Navigator Component**.
5. Click **OK**.
6. In the Name field, enter `NAV-Site Map`.
7. In the Description field, enter the following text:

   ```
   This navigator appears only on the Site Index content page.
   ```

8. In the Navigator Component section, select **Selected** from the Start Type list.

   This action indicates that you will define the site area from which the search begins. You need to define in which site to search.
9. In the Selected Start Area section, click **Select Start Area**.
10. Select **River Bend**.

Note: The site map starts its search here.

11. Click **OK**.
12. Deselect **Include Start**.
13. From the Descendant Level list, select **All**.

    This selection defines that all levels below the River Bend site are collected.
14. Deselect **Show Content**.

Note: If this option is selected, links to content in all site areas display in the navigator.

15. In the Header field, add a heading for the site map navigator by entering the following code:

```
<font class="contrastHeading">
Site Index<br>
</font>
```

16. In the Component Design 1 field, indent and use bold font for the site map navigator links by entering the following code:

```
<IndentCmpnt offset="0"
repeat="      "/>
<a href='<placeholder tag="href"/>'><b><placeholder
tag="name"/></b></a><br>
```

Note: The `IndentCmpnt` tag formats the site map into an indented hierarchy.

17. Click **Save…** and then click **Save and Close**.

7.17.2 Displaying the site map navigator as component in the Site Index content item

The site map navigator does not display on every page, as do the left navigator and the breadcrumb navigator. The site map navigator displays only on the Site Index content page. Therefore, the site map navigator is added to the PostBody field of the Site Index content item.

To display the site map navigator, do the following:

1. In the items view navigator, expand Content Library.
2. Click **Content by Title**.

Archived

3. Select the **Site Index** content item.
4. Click **Edit**.
5. In the PostBody section under Component, click **Select Component**.
6. In the items view navigator, click **Navigator Component**.
7. In the index, select **NAV-Site Map**.
8. Click **OK**.
9. Click **Save**.
10. Open the Web site and navigate to **Site Index**.

    The site map navigator is visible beneath the body text.

7.18 Creating menus

To create menus, you need to do the following:

- Create a site area menu that lists all content items that are linked to the current site are (for example, all content that is linked to the News site area).
- Reference the site area menu in the default content pages for the River Bend in the News, Bulletin Board, and Corporate Information site areas.
- Create an HTML component to format the site area menu.
- Reference the menu format in the site area's menu component design.
- Create a related information menu that lists links to related information about content pages that are created with the Three Column Table presentation template.
- Reference the related information menu in the Three Column Table presentation template.

The following sections describe these tasks in detail.

7.19 Creating a site area menu component

In this guided practice, you create the site area menu. This menu appears on the default content pages for the Colombia Coffee, African Coffee, River Bend in the News, Bulletin Board, and Corporate Information site areas.

To create a site area menu, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.

Archived

3. Click **New**.
4. Select **Menu Component**.
5. Click **OK**.
6. In the Name field, enter `MENU-Site Area`.
7. In the Description field, enter the following text:

   ```
   This menu lists the content.
   ```

8. Deselect **Include All Ancestors**.

   If this option is selected, content with parent site areas, such as Home, are found by the menu.

9. Deselect **Include All Descendants**.

   If this option is selected, content with child site areas are found by the menu.

10. Do not select an authoring template.

> **Note:** If you wanted to select documents created with a specific authoring template, you would click Select Authoring Templates. If you do not select any authoring template, all documents that are created from all available authoring templates is selected.

11. In the Menu Component section, select **Include current Site Area.**

    All content with the same site area as the currently displayed content (the section home page) matches this search criterion.

12. Be sure that the **Ascending order for results** option is selected (default).
13. In the Results Primary Sort Key list, select **Publish Date**.

    The content that is listed in this menu is sorted in order by the date on which it was published, with the latest content at the top of the list.

14. In the Results Secondary Sort Key list, select **Name**.

    If multiple pages of content have the same published date, then they are sorted alphabetically by name.

15. In the Results Tertiary Sort Key list, select **Description**, if it is not already selected as the default selection.
16. In the **Results Per Page** field, enter `0`, which means that the menu list is unlimited.
17. In the Start Page field, enter `1`, if it is not already entered as the default value.

Archived

18. In the Component Design for each matching Content field, enter the following line of code:

```
<placeholder tag="namelink"/><br>
```

19. Click **Save…** and then click **Save and Close.**

7.19.1 Referencing site area menu in specified content pages

To reference the site area menu in the River Bend in the Colombia Coffee, African Coffee, News, Bulletin Board, and Corporate Information content pages, do the following:

1. In the items view navigator, expand Content Library.
2. Click **Content by Title**.
3. Select the **River Bend in the News** content item.
4. Click **Edit**.
5. In the PostBody section under Component, click **Select Component**.
6. In the items view navigator, click **Menu Component**.
7. In the index, click **MENU-Site Area**.
8. Click **OK**.
9. Click **Save**.
10. Navigate your browser to the River Bend in the News page.

> **Note:** There is a link to the content page below the body text [Read ...]. If your menu is squeezed to the body text without separation, edit the default content for the page and press Enter to ensure that the paragraph tags are put in place.

11. Repeat steps 3 through 9 to reference the site area menu in the Colombia Coffee, African Coffee, News, Bulletin Board, and Corporate Information content pages, which are the default content pages for those site areas.

> **Note:** Before the menu can display content, create new content for those site areas. Default content is not displayed by the menu.

Archived

7.19.2 Creating an HTML component to format the site area menu

To create the HTML component that is used to format the site area menu, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **HTML Component**.
5. Click **OK**.
6. In the Name field, enter `HTML-Layout-Standard Menu`.
7. In the Description field, enter the following text:

   ```
   This component formats the site area menu.
   ```

8. In the HTML Component section, enter the following code in the field:

   ```
   <table cellpadding="0">
   <tr>
   <td>
   <b><AptrixCmpnt context="autoFill" key="Display Title" type="content"/>
   </td>
   </tr>
   <td>
   <WorkflowCmpnt context="autoFill" field="publishdate" format="EEE, MMM
   d, yyyy" type="content"/> - <AptrixCmpnt context="autoFill"
   key="Summary" type="content"/>
   <a  href='<placeholder tag="href"/>'> [Read...]</a>
   </td>
   </tr>
   </table>
   ```

Note: Autofill specifies that the content changes with each new item in the menu and is an essential definition when defining a menu.

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

7.19.3 Referencing the menu format in the site area menu's component design

To reference the format in the menu's component design, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.

3. Click **Menu Component**.
4. Select **MENU-Site Area**.
5. Click **Edit**.
6. In the Menu Component section, delete the existing text in the Component Design for each matching Content field.
7. Reference the HTML component HTML-Layout-Standard Menu by entering the following line of code in the Component Design for each matching Content field:

```
<AptrixLibCmpnt name="HTML-Layout-Standard Menu"/>
```

8. Click **Save…** and then click **Save and Close**.
9. To test your work, open the River Bend Web site and navigate to the Colombia Coffee page.

   The formatted site area menu is visible beneath the body text, as shown in Figure 7-12.The Menu now displays the Title in bold as well as the publish date.
10. Preview the other default content pages that display the site area menu (River Bend in the News, Bulletin Board, and Corporate Information).

Figure 7-12 Menu formatted with HTML Layout

Attention: At this point, you have created all the essential elements of the River Bend Tea and Coffee Company Web site. The remaining sections of this chapter show how to enhance the Web site.

7.20 Creating a menu using categories

In this guided practice, you create a menu that uses categories as search criteria. It allows users to find content throughout the Web site independently of a site area. The menu concept allows the combination of categories and site areas as search criteria.

The menu that you create in this section appears on the Espresso content page. This content page displays content that has the category *Espresso* assigned to it but that is linked to the Colombia Coffee site area. Therefore, the content is displayed by two different menus using different search criteria.

To create a menu using categories as search criteria, follow these steps:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **Menu Component**.
4. In the index, click **MENU-Site Area**.
5. Click **Edit**.
6. Click **Save…** and then click **Save As**.
7. In the Name field, enter `MENU-Category`.
8. Click **OK**.
9. In the Matching Content associated Categories section, click **Select Categories**.
10. Click Add.
11. Expand River Bend Web Site Categories and then expand Products.
12. Select **Espresso**.
13. Click **OK**.
14. Click **OK** again.
15. Deselect **Include current Site Area**.
16. Click **Save...** and then click **Save and Close**.

7.20.1 Referencing the category menu in the Espresso site area

To reference the format in the menu's component design, do the following:

1. In the items view navigator, expand Site Management.
2. Expand Site Framework.
3. Expand River Bend → Products → Coffee.

Archived

4. In the index, click **Expresso**.
5. Click **Edit**.
6. Click **Component Manager**.
7. Click **Add**.
8. In the Name field, enter `Menu`.
9. In the component reference list, select **Component Reference**.
10. Click **OK**.
11. Click **OK** again.
12. In the MENU section under Component, click **Select Component**.
13. In the items view navigator, click **Menu Component**.
14. In the index, click **MENU-Category**.
15. Click **OK**.
16. Click **Save...** and then click **Save and Close**.

7.20.2 Referencing the category menu in the Three Column Table presentation template

To reference the category menu in the Three Column Table presentation template, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select the **Three Column Table** presentation template.
4. Click **Edit**.
5. In the Presentation Template section, find the following comment in the code:

   ```
   <!-- insert related category menu component reference here -->
   ```

   Enter the following line of code beneath the comment:

   ```
   <AptrixCmpnt context="current" key="Menu" type="sitearea"/>
   ```

6. Click **Save…** and then click **Save and Close**.
7. To test your work, open the River Bend Web site in your browser and navigate to the Espresso page. The Rich Colombia Coffee from West Coast Plantation content reference displays in the menu area under the coffee bean image.

Archived

7.21 Creating the related information menu

In this guided practice, you create a related information menu. This menu appears on the content pages that use the River Bend Development authoring template and the Three Column table presentation template.

To create a related information menu, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **Menu Component**.
5. Click **OK**.
6. In the Name field, enter `MENU-Related Information`.
7. In the Description field, enter the following text:

   ```
   This menu appears in the third column on content pages created with the
   Three Column Table presentation template.
   ```

> **Note:** It does not make sense to have this menu appear in the Tools, Site Index, or Login pages, which were created with the Two Column Table presentation template. This design also applies to the Search page, which you created with the Two Column Table Search presentation template.

8. In the Menu Component section, be sure the following are selected:
   - Include all Ancestors
   - Include all Descendants
9. In the Matching Authoring Templates section, click **Select Authoring Templates**.
10. Click **Add**.
11. Select the **River Bend Development** authoring template.

    This action specifies that the menu should find content that was created using the River Bend Development authoring template.
12. Click **OK**.
13. Click **OK** again.
14. In the Matching Content associated Categories section, click **Select Category**.
15. Click **Add**.
16. Expand River Bend Web Site Categories → Corporate →News.

Archived

17. Select **External** and **Internal**.

    All content with one of the defined categories matches this search criterion.

18. Click **OK**.

19. Click **OK** again.

20. Select **Ascending order for results**, if it is not already selected.

21. In the Results Primary Sort Key list, select **Last Modified Date**.

    The content that is listed in this menu is sorted in the order of the date on which it was last modified, with the most recently edited content at the top of the list.

22. In the Results Secondary Sort Key list, leave the default value Publish Date.

    If multiple pages of content have the same modified date, then they are sorted by the date on which they were published.

23. In the Results Tertiary Sort Key list, select **Name**.

    If multiple pages of content have the same publish date, they will be sorted alphabetically by name.

24. In the Results Per Page field, enter `5`.

25. In the Start Page field, enter `1`, if it is not already entered as the default value.

26. In the Header field enter the following line of code:

    ```
    <font class="contrastHeading">Related Links</font><br>
    ```

27. In the Component Design for each matching Content field, enter the following line of code:

    ```
      <placeholder tag="namelink"/><br>
    ```

28. Click **Save…** and then click **Save and Close.**

7.21.1 Referencing the related information menu in the Three Column Table presentation template

To reference the related information menu in the Three Column Table presentation template, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select the **Three Column Table** presentation template.
4. Click **Edit**.

Archived

5. In the Presentation Template section, find the following comment in the code:

```
<!-- insert related information menu component reference here -->
```

   Enter the following line of code beneath the comment:

```
<AptrixLibCmpnt name="MENU-Related Information"/>
```

6. Click **Save…** and then click **Save and Close**.
7. Open the River Bend Web site in your browser.
8. The Related Links menu appears on the right side of the page. Select a related information menu link to view its content page.

7.22 Creating an edit document navigator

Selecting the Edit link that appears at the top of every Web page just after the content title opens the content item in the authoring portlet. To create the Edit link, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **Navigator Component**.
5. Click **OK**.
6. In the Name field, enter `NAV-Edit Doc`.
7. In the Description field, enter:

```
This navigator displays an Edit link for the current content object in
the Authoring Portlet
```

8. In the Navigator Component section, select **Current - Content** from the Start Type list.

   This option indicates that the navigator needs to go to the content of the currently displayed Site Area.

9. Be sure that Include Start is selected.

   This option indicates that the navigator must include the first item it finds (for example, the site area of the current content).

10. Select **None** from the Ancestor Level list.

    This option defines that you want to collect none of the parent site areas of the current site.

11. Select **Show Content**.

Archived

Note: If this option is selected, links to content in the current site area display in the navigator.

12. In the Component Design 1 field, enter the following code:

```
 - <a
href="/wps/myportal/wcmAuthoring?WCM_COMMAND=opencontent&ID=<Placeholder
tag="Idnum"/>">Edit</a>
```

    This code specifies to open the current content item in edit mode in the Authoring Portlet.

13. Click **Save…** and then click **Save and Close**.

7.22.1 Referencing the edit document navigator component in the presentation templates

To reference the breadcrumb navigator component in the presentation templates, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. In the Presentation Template section, append the following code behind the Display Title tag:

```
<AptrixCmpnt context="current" key="Display Title" type="content"/><br>
</font>
```

    Enter the following line of code after the tag:

```
<AptrixLibCmpnt name="NAV-Edit Doc"/>
```

    The result looks similar to the following:

```
<AptrixCmpnt context="current" key="Display Title"
type="content"/><AptrixLibCmpnt name="NAV-Edit Doc"/>
```

6. Click **Save…** and then click **Save and Close**.
7. Refresh your browser for the River Bend Web site to see the Edit link.
8. Repeat steps 3 through 6 to reference the edit document navigator in the Two Column Table presentation template.

Archived

7.23 Creating a tools navigator

To create a tools navigator that appears on every Web page, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **Navigator Component**.
5. Click **OK**.
6. In the Name field, enter `NAV-Tools`.
7. In the Description field, enter the following text:

   ```
   This navigator appears on every Web page, beneath the left navigator.
   ```

8. In the Navigator Component section, click **Selected** from the Start Type list.

   This action indicates that you will define the site area from which the search begins.

9. In the Selected Start Area section, click **Select Start Area**.
10. Expand the River Bend site and the Home site area. Click **Tools**.
11. Click **OK**.
12. Deselect **Include Start**.

> **Note:** The Tools item that appears in the navigator is not a site area. You need to enter the site area in the Header field as a non-linkable value.

13. Select **1 Level** from the Descendant Level list.
14. Deselect **Show Content**.
15. In the Header field, enter the following code for the tools navigator:

```
<form name="jump">
<font class="contrastHeading"/>
Navigation Tools
</font>
<select name="tool" size="1" style="width:140"
onchange="location.href=jump.tool.options[selectedIndex].value">
<option value="#">----- Select a tool -------</option>
```

16. In the Footer field, enter the following lines of HTML code:

```
</select></form>
```

Archived

17. In the Component Design 1 field, enter the following code:

```
<option value=<placeholder tag="href"/>> <placeholder
tag="name"/></option>
```

18. Click **Save….** and then click **Save and Close**.

7.23.1 Referencing the tools navigator component in the presentation templates

To reference the tools navigator component in the presentation templates, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. In the Presentation Templates section, find the following comment:

   ```
   <!--insert tools navigator component reference here-->
   ```

   Enter the following line of code:

   ```
   <AptrixLibCmpnt name="NAV-Tools"/><br>
   ```

6. Click **Save….** and then click **Save and Close**.
7. Refresh your browser to view the Tools navigator on the River Bend Home content page.
8. Click each link to view the default content for each site area.
9. Repeat steps 3 through 6 to reference the tools navigator component in the Two Column Table and Two Column Table Search presentation templates.

Archived

7.24 Enabling the Search module

To implement the IBM Workplace Web Content Management Search module, you need to do the following:

- Edit the connect.cfg file (Administrator task).
- Edit the aptrixsearch.properties file (Administrator task).
- Restart the Portal server (Administrator task).
- Update the Two Column Table Search presentation template to use the connect tag.
- Create a search results template.
- Create an HTML Search Form.
- Reference the HTML Search Form in the PreBody component section of the Search content item.
- Specify default content for the Search site area.

In addition to the search page, an HTML Quick Search form is created and made available on every page.

7.24.1 Editing the connect.cfg file

To edit the connect.cfg file, follow these steps:

1. Open the connect.cfg file in a text editor such as Notepad. You can find this file in the PortalServer/wcm/config directory.
2. Locate the `<Module>` section of the connect.cfg file.

Tip: You can use the text editor's search feature to locate this section quickly.

3. Locate the following tags in the `<Module>` section:

```
<!--
Aptrix Search Module
ensure that ModuleConfig/AJPE/DependentModules/AptrixSearch is
uncommented
-->
<!--
<AptrixSearch class=com.aptrix.search.AptrixSearchModule
remoteAccess=true autoLoad=false />
-->
```

Archived

4. Remove the comment tags from the `<AptrixSearch>` tag as shown in the following:

```
<!--
Aptrix Search Module
ensure that ModuleConfig/AJPE/DependentModules/AptrixSearch is
uncommented
-->
<AptrixSearch class=com.aptrix.search.AptrixSearchModule
remoteAccess=true autoLoad=false />
```

5. Locate the `<DependentModules>` section.

Tip: You can use the text editor's search feature to locate this section quickly.

6. Locate the `<AptrixSearch>` tag in the `<Dependent Modules>` section:

```
<!-- Uncomment when AptrixSearch is enabled -->
<!-- <AptrixSearch /> -->
```

7. Remove the comment tags from the `<AptrixSearch>` tag as shown in the following:

```
<!-- Uncomment when AptrixSearch is enabled -->
<AptrixSearch />
```

8. Locate the `<ProcessUnknownHosts>` tag in the `<Business Logic>` section.

Tip: You can use the text editor's search feature to locate this section quickly.

9. Change the value of this tag from `false` to `true` as shown in the following:

```
<ProcessUnknownHosts value=true />
```

10. Save and close the connect.cfg file.

7.24.2 Editing the aptrixsearch.properties file

You need to specify the user name that the search indexing program uses. At this point in our development, only the IBM Workplace Web Content Management administrator has access to the site that is you are developing. If you do not specify this user name, the indexing program does not have sufficient authority to read any content.

To edit the aptrixsearch.properties file, do the following:

1. Open the aptrixsearch.properties file in a text editor such as Notepad. You can find this file in the PortalServer/wcm/config directory.

Archived

2. Locate the following line of code:

```
#ajpepageserver.user.name=Administrator
```

3. Remove the number sign (#) and specify the `cn` attribute of the IBM Workplace Web Content Management administrator as the property value. For example, if the `cn` of the IBM Workplace Web Content Management administrator is `wcm admin`, your code looks similar to the following:

```
ajpepageserver.user.name=wcm admin
```

4. Save and close the aptrixsearch.properties file.

Important: You need to restart the Portal Server to apply the configuration changes.

7.24.3 Modifying a presentation template for the Search page

This section explains how to create a presentation template to be used with the Search page. You can use the existing Two Column Table presentation template as a starting point for creating the new template. The search results are displayed at the location of the `<connect>` tag and are formatted according to the template file that is specified in the template attribute. The PreBody component in the template is the source of the search form when you create the Search content object.

To modify a presentation template for the Search page, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. In the index, select the **Two Column Table Search** presentation template.
4. Click **Edit**.
5. In the Presentation Template section, find the following comment:

```
<!-- insert Search Results title here -->
```

6. Insert the following code beneath the comment:

```
<font class="contrastHeading">
Search Results
</font>
```

7. Beneath the Search Results title code, find the following comment:

```
<!-- format search results according to template specified in template
attribute of connect tag below -->
```

8. Beneath the comment, insert the following code for the connect tag used to obtain search results:

```
<connect mod="template" srv="search" template='<PathCmpnt
type="base"/><PathCmpnt type="context"/>/html/ResultsTemplate.html'>
</connect>
```

> **Note:** Be sure that the following PostBody component tag appears beneath the closing connect tag:
>
> ```
> <AptrixCmpnt context="current" key="PostBody" type="content"/>
> ```

9. Select **Process Connect Tags**.
10. Click **Save…** and then click **Save and Close.**

7.24.4 Creating a search results template

You need to create an HTML template that is referenced by the template attribute of the `<connect>` tag in the Two Column Table Search presentation template. The search results are formatted according to this template. To create a search results template, do the following:

1. Open a text editor such as Notepad.
2. Enter the following:

```
<html>
<body>
<resultset>
<empty>
<p>No results returned</p>
</empty>
<notempty>
<p>Results returned: <field name="resultsetsize"></p>
<table border="0">
<loop>
<tr>
   <td>
   <field name="Title"> <a href="<field
name=URL>"> [Read…]</a>
   </td>
</tr>
</loop>
</table>
</notempty>
</resultset>
</body>
</html>
```

3. Click **File** → **Save**.
4. In the Save In field, navigate to the following directory:

   Appserver/installedApps/<node>/wcm.ear/ilwwcm.war/html

   If there is no html directory, create one.
5. Enter `ResultsTemplate.html` in the File name field.
6. Save and close the file.

7.24.5 Creating an HTML search form

You need to create a search form HTML component to enable the user to enter a search query. This component is included as a PreBody component in the Search content item. To create an HTML search form, do the following:

1. In the items view navigator, click Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **HTML Component**.
5. Click **OK**.
6. In the Name field, enter `HTML-Search Form`.
7. In the Description field, enter:

   `Site visitors use this form to search the Web site from the Search page.`
8. In the HTML Component field, enter the code shown in Example 7-7.

Example 7-7 Code to create an HTML search form

```
<font class="contrastHeading">
Advanced Search
</font>
<form method="post" name="Search" action='<PathCmpnt type="base"/><PathCmpnt
type="servlet"/>/RiverBend/Home/Tools/Search/Search'>
<table border="0">
<tr>
<td>
Search Terms:
</td>
<td>
<input type="text" style="width:300px" size="120" name="QUERY">
</td>
</tr>
<tr>
<td>
Search Type:
```

```
</td>
<td>
<input type="radio" name="SEARCHTYPE" value="AND" checked="yes">All Terms (AND)
</td>
</tr>
<tr>
<td> </td>
<td>
<input type="radio" name="SEARCHTYPE" value="OR">Any Term (OR)
</td>
</tr>
<tr>
<td> </td>
<td>
<input type="submit" value="Search">
</td>
</tr>
</table>
</form>
```

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

7.24.6 Referencing the HTML search form in the Search content item

You need to reference the search form in the Search content page. You include the search form that you created as an HTML component as a PreBody component reference in the Search content page. To reference the HTML search form, do the following:

1. In the items view navigator, click Content Library.
2. Click **Content by Title**.
3. Select **Search**.
4. Click **Edit**.
5. Click **Select Component** next to the PreBody Component section.
6. In the items view navigator, click **HTML Component.**
7. In the index, click **HTML-Search Form**.
8. Click **OK**.
9. Click **Save**.
10. Refresh the River Bend Web site in a browser.
11. In the Search Terms field, enter `IBM` and click **Search**.
12. At the end of the returned result, click the **[Read…]** link.

Archived

The content displays as search result. Clicking the Read link opens the document to read the entire text.

> **Note:** The fact that the IBM Workplace Web Content Management search module returns a URL in the link in addition to the page name is a good illustration of why it is important to name each of your Web pages descriptively.

7.24.7 Creating an HTML quick search form

You need to create a quick search form so that the user can enter a search query from any Web page, not just the Search page. This component is referenced in the presentation templates. To create an HTML quick search form, do the following:

1. In the items view navigator, click Design and Development.
2. Click **Component Library**.
3. Click **New**.
4. Select **HTML Component**.
5. Click **OK**.
6. In the Name field, enter `HTML-Quick Search Form`.
7. In the Description field, enter:

   ```
   Site visitors use this form to enter search queries from any content
   page.
   ```

8. In the HTML Component field, enter the following code:

   ```
   <form method="post" name="Search"
   action='<PathCmpnt type="base"/><PathCmpnt
   type="servlet"/>/RiverBend/Home/Tools/Search/Search'>
   <font class="contrastHeading">
   Search<br>
   </font>
   <input type="text" name="query" size="15">
   <input type="submit" value="Go">
   </form>
   ```

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

Archived

7.24.8 Referencing the HTML quick search form

To reference the HTML quick search form in the presentation templates, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select **Three Column Table**.
4. Click **Edit**.
5. In the Presentation Templates section, find the following comment:

   ```
   <!- - insert quick search form html component reference here -->
   ```

   Enter the following line of code:

   ```
   <AptrixLibCmpnt name="HTML-Quick Search Form"/>
   ```

6. Save and close the presentation template.
7. Refresh your browser to view the Quick Search Form on the River Bend Home content page.
8. Click each link to view the default content for each site area.
9. Repeat steps 3 through 6 to reference the HTML quick search form in the Two Column Table and Two Column Table Search presentation templates.

7.25 Viewing content in a Portlet

Thus far, you have created the River Bend Web site as a stand-alone version without a Portal layout. This section explains how to display the Web site in a Portlet. To view content in a Portlet, you need to do the following:

- Create a Portal Page Layout presentation template.
- Create the Content Viewer Portlets.
- Create a Portal Page and add the Content Viewer Portlets.
- Configure the Portlets.

7.25.1 Creating a Portal Page Layout presentation template

The Portal Page Layout does not use a stylesheet or graphics but is kept very simple. To create the Portal Page Layout presentation template, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **New**.
3. Select **Presentation Template**.

4. Click **OK**.
5. In the Name field, enter `PortaPageLayout`.
6. In the Presentation Template section, replace the existing code with the code shown in Example 7-8.

Example 7-8 Portal page layout example code

```
<!-- page body table -->
<table width=100% cellpadding=0 cellspacing=0 border=0 class="middleBackground"
>
  <tr>
    <!-- left margin column-->
    <td align=left valign=top width=20> <AptrixLibCmpnt 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"> <AptrixLibCmpnt 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"> <AptrixLibCmpnt name="NAV-Breadcrumbs"/> </td>
        </tr>
        <tr>
          <Td> <font Class="contrastHeading"><AptrixCmpnt context="current"
key="Display Title" type="content"/><AptrixLibCmpnt name="NAV-Edit
Doc"/></font>
          </td>
        </tr>
        <Tr>
          <td>             <br>
            <AptrixCmpnt context="current" key="Body" type="content"/>
            <br>
            <AptrixCmpnt context="current" end="&lt;/li&gt;" key="content
image" type="content"/>
          <br>
          <AptrixCmpnt context="current" key="PostBody" type="content"/>
          </td>
        </tr>
        <tr>
          <Td> <AptrixLibCmpnt end="' width='375' height='1'&gt;" name="1pixel"
start="&lt;img src='"/>
          </td>
```

```
        </tr>
      </table>
    </td>
    <!-- middle gutter -->
    <td valign=top align=left width=20> <AptrixLibCmpnt end="' width='20'
height='1'&gt;" name="1pixel" start="&lt;img src='"/>
    </td>
    <!-- aux content col -->
    <!-- right margin column -->
    <td valign=top align=left width=10 > <AptrixLibCmpnt end="' width='10'
height='1'&gt;" name="1pixel" start="&lt;img src='"/>
    </td>
  </tr>
</table>
```

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

7.25.2 Creating and copying Content Viewer Portlets

The Local Rendering Portlet is installed with the Authoring user interface. You make copies of this user interface. You are specifying your own configuration for the River Bend Web site, and you do not to configure the base portlet. To create two copies of the Web Content Viewer portlet, do the following:

1. Open WebSphere Portal Administration by clicking **Administration** on the top right navigator in your browser window, as shown in Figure 7-13.

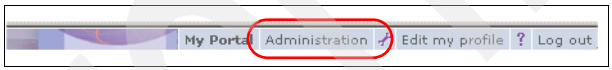


Figure 7-13 WebSphere Portal Administration

2. Click **Portlet Management**.
3. Click **Portlets**.
4. In the Search window, type `Web` and then click **Search**.
5. Find the Web Content Viewer portlet and click the **Copy Portlet** icon.
6. In both the portlet application and the portlet fields, enter the new name:

   `WCM River Bend Navigator`

7. Create a second portlet application and portlet by clicking the **Copy Portlet** icon again.
8. In both the portlet application and the portlet fields, enter the new name:

   `WCM River Bend`

9. Click **OK**.

The new Portlets are listed in the Portlets list as shown in Figure 7-14.

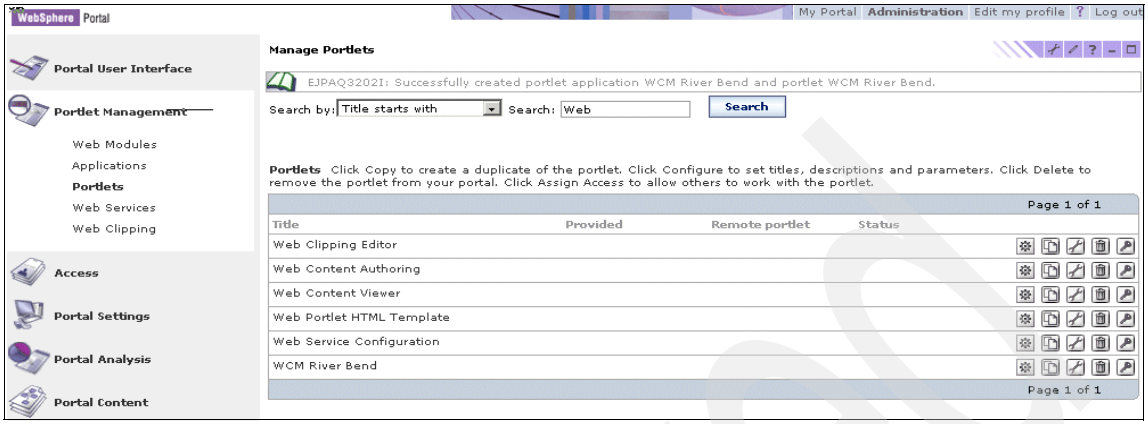


Figure 7-14 River Bend Portlets in Portlets List

7.25.3 Creating a Portal Page and adding the Content Viewer Portlets

To create a Portal Page and to add two Portlets to the page, do the following:

1. Click **Portal User Interface**.
2. Click **Manage Pages.**
3. Click the **My Portal** link in the Pages in content Root list.
4. Click New Page.
5. In the Title field, enter `River Bend Coffee and Tea`.
6. In the Type of Page section make sure in the two column property is selected (default value).
7. Click **OK**.

   The following message appears under Page Properties:

   `River Bend Tea and Coffee has been created successfully.`

8. Click **OK**.
9. Using the up arrow on the right of Status column in the River Bend Tea and Coffee page link row, move it up in the list until it is under IBM Workplace Web Content Management.
10. Select the **Edit Page Layout** icon.
11. In the left column of the window, click **Add Portlets**.
12. In the Search window, type `WCM` and click **Search**.
13. In the returned portlet list, select **WCM River Bend Navigator**.

Archived

14. In the right column, click **Add Portlets**.
15. In the Search window, type `WCM` and click **Search**.
16. Select **WCM River Bend**.
17. Click **OK**.
18. Click **Done**.

The newly created Portlets are added to a new Portal Page, as shown in Figure 7-15.

Figure 7-15 River Bend Portlets on River Bend Page

7.25.4 Configuring Portlets

You need to configure the new Portlets on the River Bend Coffee and Tea Portal Page to show the NAV-Left navigator in the left Portlet and content in the right Portlet. To configure the Portlets, follow these steps:

1. Select **My Portal** from the top navigation
2. Select the new **River Bend Coffee and Tea** page.
3. In the left Portlet, click the **Configuration** icon, as shown in Figure 7-16.

Figure 7-16 Configuration icon

4. In the Content section, select **Library Component**.
5. Click **Edit** in the Library Component section.

6. In the content item navigator, select **Navigator Component**.
7. Select **NAV-Left**.
8. Click **OK**.
9. Click the Edit icon in the Default Context section.
10. In the content item navigator, expand **River Bend**.
11. Select **Products**.
12. Click **OK**.
13. In the Links section, select **This page** under Broadcast Links To.
14. Under Receive Links From, select **Other portlets and this portlet**.
15. Click **OK**.

The Portlets are configured, as shown in Figure 7-17.

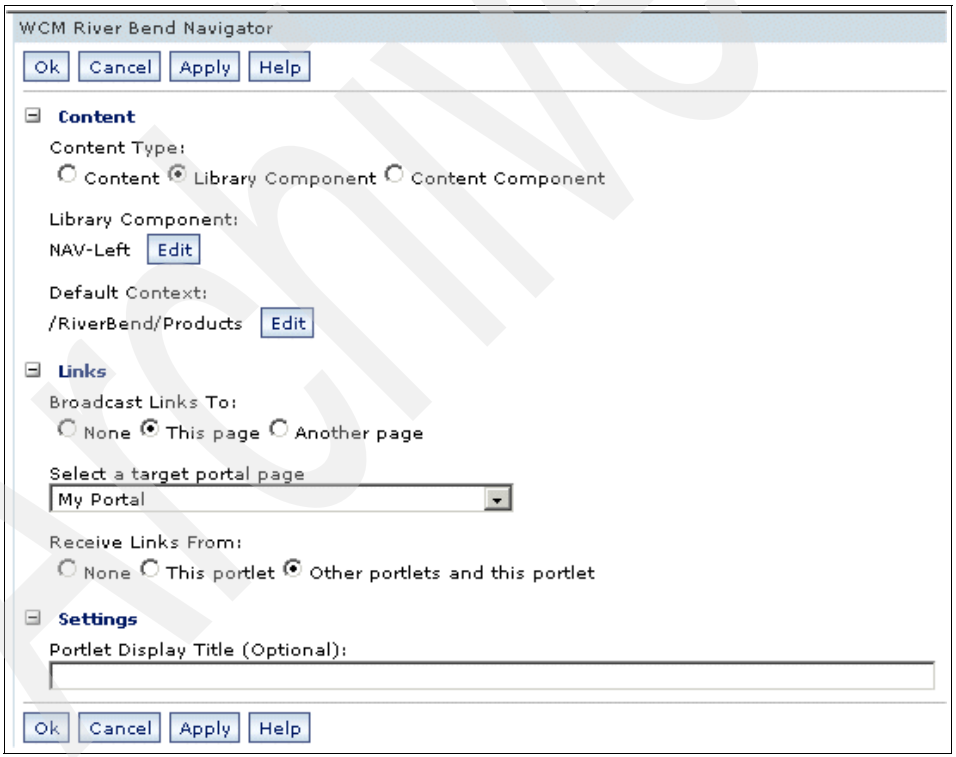


Figure 7-17 Configuration for Portlet NAV-Left

16. In the right Portlet click the **Configuration** icon.
17. In the Content section, select **Content**.

18. Click **Edit** under Content.
19. In the content item navigator, expand **River Bend**.
20. Select **Products**.
21. Click **OK**.
22. Click **Edit** under Alternate Presentation Template.
23. Select **PortalPageLayout**.
24. Click **OK**.
25. In the Links section, select **This page** under Broadcast Links To.
26. Under Receive Links From, select **Other portlets and this portlet**
27. Click **OK**.

The left Portlet shows the navigator, and the right portlet shows the Product content page, as shown in Figure 7-18.

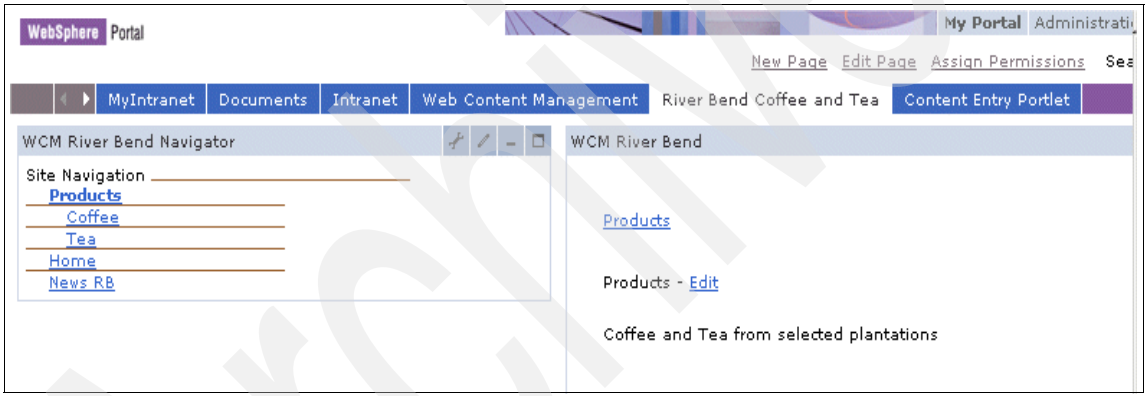


Figure 7-18 River Bend Web site displayed in a Portal page

Note: The Portal configuration is very basic and does not cover all Components that you have created in this chapter (for example, Menus or Images are not referenced in the PortalPageLayout.). This chapter is only a first step into page creation.

Archived

7.26 Integrating Portal Document Manager with IBM Workplace Web Content Management 5.1

Portal Document Manager integration is available only for IBM Workplace Web Content Management 5.1. There is no installation necessary. However, to use a Portal Document Manager document within IBM Workplace Web Content Management, you need to do the following:

- In Portal Document Manager, you need to:
  - Create a Library.
  - Create a Folder.
  - Create Documents.
- In IBM Workplace Web Content Management, you need to:
  - Create an authoring template with a Portal Document Manager component. The fields in the new authoring template will match the existing presentation template component references.
  - Create content with the new Portal Document Manager authoring template.

This section explains how to integrate Portal Document Manager documents into BM Workplace Web Content Management 5.1.

7.26.1 Configuring and creating library, folder, and documents in Portal Document Manager

You create the library, provide access to the library, and configure a Portal page to display the library through an administrator task using the Administration interface. To create the folders and documents in the library is a user task that is performed using the MyPortal GUI.

Configuring and creating a library

To configure a library in Portal Document Manager, you use WebSphere Portal Administration. Follow these steps:

1. Open WebSphere Portal Administration by clicking **Administration**.
2. On the left side, click **Portal Content**.
3. Click **Manage Document Libraries**.

   By default, the Document Manager library is created. If you wish to create a new library, click **New Document Library** and complete the required fields.

4. Click the **Edit** Icon.
5. In the Versioning section, select **Enable document versioning**.

> **Note:** To update your IBM Workplace Web Content Management content with new versions of the Portal Document Manager document, you need to version the Portal Document Manager documents and use this feature in the IBM Workplace Web Content Management content object.

6. Keep all other default values and click **OK**.

Creating a folder

> **Note:** Only one library per page is displayed in WebSphere Portal Server. You must use the Administration console to change the library that is displayed for a page.

To create a folder in Portal Document Manager, follow these steps:

1. Click **My Portal**.
2. Click the **Documents Portal** page.
3. Click **New**.
4. Select **Folder**.
5. In the Folder **name** field, enter: `WCM River Bend`.
6. Click **OK**.

The IBM Workplace Web Content Management River Bend folder is displayed on the left side in the library navigation.

Creating a document

To create content for River Bend in Portal Document Manager, follow these steps:

1. Open the WCM River Bend folder.
2. Click **Import File**.
3. In the Title field, enter: `Redbook Team News`.
4. Click **Browse** and select a Microsoft Word file from your hard drive.
5. Click **Open.**
6. Click **Publish**.

Archived

Restriction: In IBM Workplace Web Content Management 5.1, there are known issues with special characters. Special characters (such as letters that contain symbols — for example, ö or ä) are not rendered correctly when a Portal Document Manager document that contains these characters is published within IBM Workplace Web Content Management. Refer to the IBM support Web site to download the iFix PK05525 to your installation, which is available at:

http://www-128.ibm.com/developerworks/workplace/support/downloads.html

7.26.2 Creating an authoring template with a Portal Document Manager component

During this exercise, you create a new authoring template with a Portal Document Manager component. To match the component references in the existing presentation templates, the fields are identical — despite the Portal Document Manager component.

To create a copy of the River Bend authoring template and to add a Portal Document Manager component, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Authoring Templates**.
3. Click **River Bend Development**.
4. Click **Edit**.
5. Click **Save...** and then click **Save As**.
6. In the New Name field, type `River Bend Portal Document Manager`.
7. Click **OK**.
8. Click **Component Manager**.
9. Select the following and then click **Remove**:
   - Prebody
   - Postbody
   - Content Image
10. Click **Add**.
11. In the Name field, type `PDMFile`.
12. In the Component Reference list, select **Document Manager Component.**
13. Click **OK**.
14. Click **OK** again.

Archived

15. Click **Save…** and then click **Save and Close**.

7.26.3 Referencing the new component in the presentation template

To reference the Portal Document Manager Component in the presentation templates, do the following:

1. In the items view navigator, expand Design and Development.
2. Click **Presentation Templates**.
3. Select the **Three Column Table** presentation template.
4. Click **Edit**.
5. In the Presentation Template section, find the following:

   ```
   <AptrixCmpnt context="current" key="PostBody" type="content"/>
   ```

   Enter the following line of code:

   ```
   <DocumentManagerCmpnt context="current" convertTo="text/html"
   key="PDMFile" scope="Document" type="content"/>
   ```

6. Save and close the presentation template.
7. Repeat steps 3 through 6 to reference the Two Column Table and the PortalPageLayout presentation template.

7.26.4 Pairing the Portal Document Manager authoring template and the presentation template

To pair the authoring templates and presentation templates appropriately for the River Bend site and its various site areas do the following:

1. In the items view navigator, expand Site Management.
2. Expand Site Framework.
3. Select **River Bend**.
4. Click **Edit**.
5. In the Properties section, click **Edit Template Mapping**.
6. Click **Edit**.
7. In the Authoring Templates section, select the River Bend Portal Document Manager authoring template.
8. In the Presentation Templates section, click the Three Column Table presentation template.

Archived

9. Click **OK**.

   The two templates are displayed as mapped.

10. Click **OK**.

11. Click **Save…** and then click **Save and Close**.

7.26.5 Creating content using the River Bend Portal Document Manager authoring template

To create a content item with the new authoring template using the Portal Document Manager integration, do the following:

1. In the items view navigator, expand Content Library.
2. Click **New**.
3. Select **Content**.
4. Click **OK**.
5. Select the River Bend Portal Document Manager authoring template.
6. In the Site Framework view in the items view navigator, expand the River Bend site and News site area.
7. Click **River Bend in the News**.
8. Click **OK**.
9. In the **Name** field, enter `WCM Redbook Team News`.
10. Expand the Profile section, and click **Select Categories**.
11. Click **Add**, and expand the following:
    - River Bend Web Site Categories taxonomy
    - Corporate category
    - News category
12. Select **Internal** and **External**.
13. Click **OK**.
14. Click **OK** again.
15. In the Display Title field, enter `WCM Redbook Team News`.
16. In the Summary field, enter `Recent news about WCM Redbook Team`.
17. Clear the Body field.
18. Scroll to the PDMFile section.
19. In the PDMFile section, click **Search**.

Archived

20. In the new window, expand Document Manager → WCM River Bend, and click **Redbook Team News**.

> **Note:** The file name and title appear in the corresponding fields.

21. Click **OK**.

> **Note:** The path fields in the authoring template show path and file name.

22. Click **Save…** and then click **Save and Close**.
23. To assign this content item as the default content page for the River Bend in the News site area, expand Site Management in the item views navigator.
24. Expand Site Framework.
25. Expand the River Bend site and the News site area.
26. Select **River Bend in the News**.
27. Click **Edit**.
28. In the Properties section under Default Content, click **Select Default Content**.
29. Select the **River Bend in the News** content page.
30. Click **OK**.
31. Click **Save…** and then click **Save and Close**.
32. Open the River Bend Web site with your browser. Navigate to River Bend in the News and find WCM Redbook Team News which displays the Portal Document Manager document content.

Archived

8

Caching and pre-rendering considerations

This chapter discusses performance considerations in IBM Workplace Web Content Management through the use of pre-rendering and caching strategies. It includes the following sections:

- Improving performance
- Caching options
- Pre-rendering options
- Implementing a caching or pre-rendering strategy
- Setting up caching
- Configuring data caching
- Configuring pre-rendering
- Custom caching with connect tags
- Considerations in developing a caching strategy
- Documenting the cache strategy
- WebSphere Dynamic Cache Service

© Copyright IBM Corp. 2006. All rights reserved.

8.1 Improving performance

A typical IBM Workplace Web Content Management implementation includes dynamically generated pages that often combine navigation elements with files (such as stylesheets or JavaScript™ scripts), images, and site content. Each of these components can have special security settings. As well, menus can be designed to show different content based on a user's profile.

While this flexibility is one of the many strength's of IBM Workplace Web Content Management, it can also lead to reduced performance. Consequently, it is important to analyze performance carefully and to implement tuning properly in any IBM Workplace Web Content Management implementation.

Because IBM Workplace Web Content Management runs as part of a WebSphere Portal environment, its performance is dependent on many settings that are outside of the IBM Workplace Web Content Management configuration. For example, IBM Workplace Web Content Management performance depends on hardware, Web server caching, WebSphere Application Server settings, and WebSphere Portal Server settings.

IBM Workplace Web Content Management offers caching and pre-rending as solutions for improving performance within the IBM Workplace Web Content Management framework. Additionally, you can use *dynamic cache* to improve performance. By default, dynamic cache (via WebSphere Dynamic Cache Service) is enabled on an application server. It supports caching of servlet and JSP responses, WebSphere commands objects, Web services objects, and Java objects.

When an IBM Workplace Web Content Management server is configured to cache or pre-render content, it stores partial or complete copies of dynamically generated pages. By leveraging this stored content, the IBM Workplace Web Content Management server can deliver pages more quickly, because it can omit the page assembling process partially or completely.

Caching and page rendering differ in that *cached content* is added to the cache on an *as needed basis* (that is, when it is requested by a user and when it is not in the cache). Cached content is removed from the cache when it expires or when the cache is full and more recent content needs to be cached. In contrast, *page rendering* occurs as a *batch operation*. All of the content for a site is pre-rendered at roughly the same time.

You can configure both caching and pre-rendering in such a way that allows for dynamic components within a page (for example, to accommodate components with security or profile settings or connect tags that show live data).

8.1.1 Caching options

IBM Workplace Web Content Management offers many different caching options. However, all caching in IBM Workplace Web Content Management shares the following basic characteristics:

- Web pages are cached after a user requests the Web page. This strategy has two basic effects:
  - Generally, when a Web page that has not been cached is requested, it displays more slowly than a Web page that is delivered from the cache.
  - Cached pages can have broken links. For example, it is possible for a page with a menu to show links to content that has been expired or removed recently from the site.
- Cached Web pages expire and are removed automatically from the cache based on cache expiration settings.

  IBM Workplace Web Content Management supports default expiration settings for caching throughout the site. In addition, IBM Workplace Web Content Management Advanced caching supports custom expiration settings for individual components in the site. You can adjust expiration settings to reduce the likelihood of broken links due to *stale* cached pages.

Basic caching

Basic caching is the simplest caching strategy that is available in IBM Workplace Web Content Management. Basic caching creates a single cached instance of each content object for all users — whether known or anonymous. Basic caching has the following characteristics:

- It cannot be used in sites that contain placeholders for dynamic content such as connect tags.
- It only supports sites where all content (including Library Components such as navigators and menus) are displayed identically for all site visitors.
- It does not allow connect tags (whether to IBM Workplace Web Content Management Library Components, Web pages, or database queries) to be configured to implement any other caching strategy.

Basic caching can be a successful strategy on a site where all content is static and where all content is displayed the same way for all users. Sites that are more complex should not employ basic caching.

Advanced caching

Advanced caching allows for far more flexibility than basic caching. However, this flexibility comes at a price. Configuring a site to use advanced caching can be more complicated and more time-consuming than configuring a site to use basic

Archived

caching. Also, advanced caching requires more disk space and more memory, and it can result in slower performance than basic caching. Advanced caching has the following characteristics:

- It supports dynamic content (such as content brought in with connect tags).
- It supports configuring dynamic content to use a different caching strategy from the default caching strategy. For example, if the content is the same for all users in a site, a site could be configured to use one site cache for all users. Then, any special menus that are drawn based on the user's profile could be coded to use special caching.
- It supports caching content in several different ways:
  - *Site caching*, which works much like basic caching, creates a single cache for all site visitors. Unlike basic caching, site caching supports adding caching parameters to connect tags so that parts of the site (such as menus) can be cached differently from most of the site.
  - *Session caching* creates a copy of each Web page in the session cache for each user's session. A cached page is created as the user visits the page. After the page is first cached, the user sees the cached page until starting a new session or until the page expires from the cache (whichever comes first).
  - *User caching* creates a copy of each Web page in the user cache for each user. Like session caching, a Web page is cached when the user visits a page and it expires. Unlike session caching, the Web page in the user cache lives beyond the current session.
  - *Secured caching* is used in sites that control access to Web pages and components based on the user's group memberships.
  - *Personalization caching* is used to share a cache among all users with the same *personalization profile*. Users with the same profile (categories and keywords) and group membership share a single cache.

Caching paradigms

IBM Workplace Web Content Management supports the following caching paradigms:

- *Default caching* is configured at the site level through the IBM Workplace Web Content Management configuration file (connect.cfg). It defines how any content that is not otherwise specified should be cached. In general, basic caching, site caching, and session caching are only used as your server's default Web content cache.
- *Custom caching* is configured in connect tags and URL requests. Custom caching overrides the IBM Workplace Web Content Management server's default caching strategy. Custom caching is possible only when no cache or

advanced caching (generally, site or session caching) is used as the site's default caching strategy. You can use custom caching to cache parts of a page. For example, a connect tag that shows a navigator might employ secured caching if links vary depending on a user's access. Similarly, a connect tag showing a menu might employ personalized caching if links vary depending on a user's profile.

Choosing between caching strategies

In general, basic caching is preferred over advanced caching for sites that show the same static content for all users. Advanced caching is preferred over basic caching when different users should see a site differently, when a site includes dynamic content, or when certain parts of a site need to be cached differently from most of the content in the site.

Note: Keep in mind that advanced caching requires more memory and disk space than basic caching. Also, basic caching can have much faster performance than advanced caching.

8.1.2 Pre-rendering options

IBM Workplace Web Content Management has two ways to deliver pre-rendered content: via an HTTP server or via IBM Workplace Web Content Management. In both cases, a site is pre-rendered as part of a regularly scheduled batch process. As such, pre-rendered sites are far less likely to show broken links (because all pages are rendered at roughly the same time). As well, pre-rendered sites have faster performance than cached sites on pages that have not yet been added to the cache.

Pre-rendering via an HTTP server

A site that is pre-rendered via an HTTP server is very similar to a site with basic caching, in that the content must be static and custom caching cannot be used. However, there are some significant differences between pre-rendering via an HTTP server and basic caching, as follows:

- Because all pages in a site are pre-rendered, pre-rendering generally requires more disk space than basic caching (which maintains copies of recently accessed pages).
- Because pre-rendering creates a copy of every page at roughly the same time, pre-rendering is less likely to result in broken links than basic caching (which caches pages as needed). As well, it is likely to result in higher performance, particularly when accessing less popular pages.

Table 8-1 lists the differences between a site that is pre-rendered via an HTTP server and site that is cached using basic caching.

Table 8-1 Pre-rendering via an HTTP server versus basic caching

	Basic caching	**Pre-rendered site delivered via an HTTP server**
Performance	Very fast	Extremely fast
Connect tag processing	No	No
Custom Caching	No	No
Memory Requirements	Low to Medium	Memory requirements depends on the Web server that is used.
Disk Requirements	Low to Medium	Potentially very high as the entire site must be able to fit on disk.
Unexpected Broken Links	Yes Because some pages might be cached at different times, there is a small chance that not all the links on a cached page are valid currently.	No The site is pre-rendered in a single batch, which greatly reduces the chances of inconsistencies in the site.

Pre-rendering via IBM Workplace Web Content Management

A site that is pre-rendered via IBM Workplace Web Content Management is very similar to a site with advanced caching, in that content can be dynamic. In addition, different caches can be used by different users and different caching strategies can be used for different site components.

Because pre-rendered pages are generated all at once — while cached pages are generated on an as-needed basis — pre-rendered pages are less likely to have broken links than cached pages. Web pages that are not in the cache display more slowly than pre-rendered pages.

Note: Because these options are very similar, it is generally necessary to test both strategies before deciding which is better for a site.

Table 8-2 lists the differences between a site that is pre-rendered via IBM Workplace Web Content Management and site that is cached using advanced caching.

Table 8-2 IBM Workplace Web Content Management versus advanced caching

	Advanced Caching	**Pre-rendered site delivered via IBM Workplace Web Content Management**
Performance	Fast when cached, but slower if the requested page has expired from the cache. (Because tag processing has a cost, this depends on how many connect tags a page contains.)	Fast, but because tag processing has a cost, this depends on how many connect tags a page contains.
Connect tag processing	Yes	Yes
Custom Caching	Yes	Yes
Memory Requirements	Medium to high	Medium to high
Disk Requirements	Medium to high	Medium to high
Unexpected Broken Links	Yes Because some pages might be cached at different times, there is a small chance that not all the links on a cached page are valid currently.	No The site is pre-rendered in a single batch, which greatly reduces the chances of inconsistencies in the site.

8.2 Implementing a caching or pre-rendering strategy

It is possible to develop a successful IBM Workplace Web Content Management site without implementing any caching or pre-rendering strategy. Nevertheless, it is important to test the performance of a site before rolling it to production in order to ensure that its production performance will be acceptable. As well, it is important to test a site's performance regularly, because changes to the site's design and user base can affect performance and can require changes to the caching or pre-rendering strategy.

The following steps are typically part of the successful implementation of a caching or pre-rendering strategy:

1. Develop a site.

Archived

Set up the site without implementing any caching. Ensure that it has a realistic representation of the content the site will provide, including any dynamic or personalized content.

2. Configure the environment.

   Ensure that the new site has IBM-recommended hardware as well as IBM-recommended parameters for the following:

   – WebSphere Application Server and WebSphere Portal Server tuning parameters (for IBM Workplace Web Content Management and WebSphere Portal)
   – Network properties (for WebSphere Portal)
   – Database parameters (for IBM Workplace Web Content Management and WebSphere Portal)
   – HTTP Server tuning and caching parameters (for WebSphere Portal)
   – Operating System settings (for IBM Workplace Web Content Management and WebSphere Portal)
   – WebSphere Portal Service properties (for WebSphere Portal)
   – If applicable, LDAP Server parameters (for IBM Workplace Web Content Management and WebSphere Portal)

3. Test performance for the site.

   In this step, you want to test a site that is configured properly for production for two important reasons:

   – To establish a baseline of performance.
   – To determine whether it is necessary to enable any caching strategy.

   It is highly preferable that you use an automated testing strategy so that stress-testing is measured properly. In addition, we recommend that you test ranges of suggested values in a few different ways to find the optimal settings for the environment.

   Note: If testing reveals that the performance is acceptable, you might want to skip several steps and move right to rolling the site to production. Alternatively, you can continue with the following steps, further improving performance.

4. Analyze the site's content.

   Is the content static, is it dynamic (that is, does it contain connect tags that show HTML from other Web pages, SQL result sets, or JSP components), or is it both? Are menus drawn differently based on the user's profile? Is content displayed and hidden using security?

5. Design a caching strategy.

   If the performance test revealed that a caching strategy is necessary, the analysis of the site should suggest which of the four strategies would be most appropriate.

   > **Note:** Sometimes (particularly when choosing between advanced caching and pre-rendering via Web content management), it is necessary to implement and stress-test each strategy before making a final choice.

6. Implement caching in a development.

   It can take quite a bit of time to implement a caching strategy. In addition, implementing a caching strategy can sometimes have unexpected results. It is important to first implement caching in a development environment to avoid disrupting your production environment. It is recommended that your development environment closely mirrors your production environment so that testing results are meaningful.

7. Test the cached site.

   Perform functional testing (ensure that content is cached correctly). You should also perform stress testing to ensure that performance is acceptable.

8. Make adjustments as necessary.

   Implementing a caching strategy often requires an iterative approach. So, you might need to perform steps 5, 6, and 7 until test results are satisfactory.

9. Implement caching strategy.

   Implement all changes in the production environment.

10. Expect to revisit the caching strategy.

    Healthy sites change over time. It often happens that characteristics of a pilot site (such as the absence of dynamic content) changes over time. Because sites change, it is important to keep in mind that any caching strategy might need to be updated over time as a site's content and design evolve.

Developing and implementing a caching or pre-rendering strategy requires an iterative process of designing, developing, and testing. While developing a strategy can be time-consuming, the work is rewarded with improved performance.

However, as you develop a caching or pre-rendering strategy, do not lose sight of the larger picture. Significant performance gains on IBM Workplace Web Content Management sites can often be found by using adequate hardware as well as by tuning other parts of the environment, including Web server caching, WebSphere Application Server settings, and WebSphere Portal Server settings. Configuring

Archived

the environment correctly, as discussed in step 2 on page 354, is critical to the success of an IBM Workplace Web Content Management site.

8.3 Setting up caching

Setting up IBM Workplace Web Content Management cache requires changes to the server's connect.cfg configuration file. The section provides a guide to the properties and attributes that you need to implement a cache strategy.

8.3.1 Understanding expiration parameters

All cached items have an expiration time that indicates when a cached item is eligible to be removed. An item is removed when the expiration time is reached, when the cache exceeds it memory capacity, or when a new request is made for the expired item. Expiration parameters are used to configure the expiration time. You can configure an item's expiration time to be either *relative* or *absolute*. A relative expiration time is calculated each time the cache is created. Relative expiration parameters are the most flexible and the most commonly used. Absolute expire times create a cache entry that expires at a specific time. When the absolute expiration time has passed, each time the cache is rebuilt, it expires immediately. Improper use of absolute parameters can cause unnecessary overhead, especially if the item is requested frequently.

To configure cache expiration, an absolute date and time or a relative period is set in the default cache configuration, as URL parameters and in connect tag attributes. Expiration parameters are not required on HURLs or in connect tags. If the expiration parameter is not included, the default cache configuration expiration time is used. These properties use a specific syntax for their values. Table 8-3 defines the syntax for the absolute and relative expiration settings.

Table 8-3 Syntax for absolute and relative expiration settings

Expiration type	Proper format and examples

Absolute	ABS {date and time}
	
	Date and time must conform to one of the following formats:
	`Mon., 06 Nov. 2000 09:00:00 GMT`
	`Monday, 06-Nov.-00 09:00:00 GMT`
	`Mon. Nov. 6 09:00:00 2000`
	`6 Nov. 2000 9:00 AM`
	For example:
	`ABS Mon., 29 May 2000 03:04:18 GMT`

Expiration type	Proper format and examples

Relative	REL {integer}{units}
	
	Integer indicates whole number units of time.
	Units are represented by a single case-insensitive letter, as follows:
	`S: Seconds`
	`H: Hours`
	`D: Days`
	`M: Months`
	Examples:
	► In connect tag:
	– `REL 9000s`
	– `REL 7D`
	► In a URL:
	– `REL+9000s`
	– `REL+7D`

Tip: In general, using relative time periods is preferred over absolute dates. When an absolute date has past, any future requests for content build a cache that expires immediately after it is rendered.

For more information about the cache expiration setting, see the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

8.3.2 Configuring the default cache

The first step in implementing a caching strategy is to configure the default cache. The cache configuration is divided into an IBM Workplace Web Content Management cache and a data cache. The IBM Workplace Web Content Management cache is the default cache for content that is stored within an IBM Workplace Web Content Management repository. External content, HTML, or SQL data that is referenced using connect tags is cached using the data cache. This section describes how to configure the IBM Workplace Web Content Management cache. You can find details about how to configure the data cache in 8.4, “Configuring data caching” on page 362.

Default cache is configured in the connect.cfg configuration file. The file is located in <*WPS Root*>/wcm/config directory.

- On Windows, you can find the file here:

  C:\Program Files\WebSphere\PortalServer\wcm\config\connect.cfg

- On Linux®, you can find the file here:

  /opt/WebSphere/PortalServer/wcm/config/connect.cfg

Before altering the connect.cfg file, decide if you will use basic caching, advanced caching, or pre-rendering. The properties that you modify for basic caching, advanced caching, and pre-rendering are quite different. Thus, it is important to know which cache you are implementing.

The caching properties are located throughout the connect.cfg file. Some properties are unique to a particular cache type. Other properties are repeated throughout the configuration but provide the same function. The following lists and describes those properties:

- `FlushExpiredInterval`: Controls how often the cache flushes expired items from disk and memory. Specify an integer to indicate the number of seconds between cache flushings.
- `WriteIndexInterval`: Controls how often the cache index is written to disk. Specify an integer to indicate the number of seconds. If `0` is specified, all changes are written to disk.
- `CacheDir`: The directory on the server where cache is stored.
- `MemCacheSize`: Indicates the number of items that are stored in the memory cache. If set to `0`, the cache is not created.
- `DiskCacheSize`: Indicates the number of items that are stored in the disk cache. If set to `0`, the cache is not created.

> **Note:** The connect.cfg file is formatted in a syntax that is similar to XML. To help clarify which properties are being changed, the property names in the following sections are formatted to reflect the complete path to the property. For example, the property `DefaultCache` is found in the `BussinessLogic` section. The path to the value is /BussinessLogic/DefaultCache.

8.3.3 Understanding cache size

Items are stored normally in the cache until they expire. If the cache size limit is reached, then the cache removes items using a *first-in, first-out* rule. Older items are removed first to make room for new items. If the cache size is too small, the site performance can degrade due the constant swapping. Ideally, the memory and disk cache should be large enough to allow cache to expire normally. Be aware that adjusting the cache size affects how much memory and

storage resources are needed to run IBM Workplace Web Content Management. Consult with the system's administrator before changing these values.

8.3.4 Configuring basic caching

The following properties are used to configure basic caching:

- `/BussinessLogic/DefaultCache`: Enables basic caching as the default cache. To enable basic caching set value to `true`.
- `/BussinessLogic/DefaultCacheExpires`: Sets the default cache expiration. Items remain in cache until the expiration time has past or until the cache exceeds is maximum size. Set the value using the expiration syntax as discussed in 8.3.1, “Understanding expiration parameters” on page 356.

Example 8-1 shows a basic cache configuration in the connect.cfg file.

Example 8-1 Basic cache configuration in the connect.cfg file

```
...
<BusinessLogic>
   …
   <DefaultCacheExpires value="REL 3D" />
   <DefaultCache value=true />

   <ModuleReponseCacheConfig>
      <FlushExpiredInterval value=120 />
      <WriteIndexInterval value=90 />
      <CacheDir value="../connect/module/" />
      <MemCacheSize value=1000 />
      <DiskCacheSize value=5000 />
   </ ModuleReponseCacheConfig>
...
```

These settings enable IBM Workplace Web Content Management to use basic caching as the default for managed content. Not all properties need to be adjusted to enable basic caching. The `ModuleResponseCacheConfig` properties remain their default values. Usually, the default values are sufficient to begin testing your caching strategy.

Important: When using basic caching, JSP components do not function.

8.3.5 Configuring advanced cache

Configuring advanced caching requires more configuration than basic caching. Advance caching permits default settings to be stored in the connect.cfg file and then to be overridden by including expiration and cache type attributes on connect tags. Selecting the advance cache type is the first step in the

configuration. The typical choices are *site* or *session* cache types. Review "Advanced caching" on page 349 before selecting a default cache type.

The following properties are used to configure advance caching as the default cache:

- `/ModuleConfig/AJPE/ContentCache/DefaultContentCache`: Sets the default advance cache type for the site. Typical values are `site` or `session`.
- `/ModuleConfig/AJPE/ContentCache/ContentCacheExpires`: Sets the default expiry for all advance cache. Set the value using the expiration syntax as discussed in 8.3.1, "Understanding expiration parameters" on page 356.
- `/SessionCacheConfig/MemCacheSize`: Used only for session cache. The value indicates the number of items that are stored in cache for sessions.

Example 8-2 shows an advanced cache configuration in the connect.cfg file.

Example 8-2 Advanced cache configuration in the connect.cfg file

```
...
<ModuleConfig>
   <AJPE>
   ...
      <ContentCache>
         <DefaultContentCache value="Site" />
         <ContentCacheExpires value="REL 2D" />
         <FlushExpiredInterval value=120 />
         <WriteIndexInterval value=90 />
         <CacheDir value="../ilwwcm/system/contentcache" />
         <MemCacheSize value=1000 />
         <DiskCacheSize value=5000 />
      </ContentCache>
</AJPE>
...
</ModuleConfig>
...
<SessionCacheConfig>
   <MemCacheSize value=1000 />
</SessionCacheConfig>
...
```

Example 8-2 on page 360 configures IBM Workplace Web Content Management's default cache to use *advanced site caching*. In this example, the default cache sizes and cache directory remain. While testing the cache strategy, you might need to adjust the cache sizes to meet performance requirements.

Often when implementing advanced caching, there is a need to override the default cache. Connect tags can override the default cache by including cache

type and expiration attributes on the tag instance. Before using connect tags to override the default cache, there are some configuration steps.

First, it a good practice to copy the default module property from the connect.cfg file and to give it a custom name for the connect tags to use. Using the default module works, but if you decide in the future to change the default module's implementation class, the connect tag entries might not function or might need to be re-written. In the connect.cfg file, locate the `/BusinessLogic/Module/Default` property. Copy it and rename it to `/BusinessLogic/Module/ComponentRender`.

Example 8-3 shows an example of using connect tags to override the default cache.

Example 8-3 Using connect tags to override the default cache to provide flexibility

```
...
<BusinessLogic>
...
   <Module>
...
<Default class=com.aptrix.pluto.renderer.RendererModule remoteAccess=true
autoLoad=fale />
<ComponentRenderer class=com.aptrix.pluto.renderer.RendererModule
remoteAccess=true autoLoad=fale />
...
```

Note: To verify the implementation class, refer to the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

Depending on which version of IBM Workplace Web Content Management you are using, the implementation class name might begin with `com.ibm.workplace` instead of the existing Aptrix class names.

Second, by default IBM Workplace Web Content Management only allows a connect tag to be processed by known hosts. If IBM Workplace Web Content Management is accessed via an IP address or multiple host names, you need to update the configuration. There are two approaches to this issue. You can either disable this feature, or you can update the IBM Workplace Web Content Management configuration and list all applicable hosts and IP addresses. The excerpt shown in Figure 8-1 identifies the properties.

Archived

```
...
<BusinessLogic>
      <ProcessUnknownHosts value=false />
      <Hosts>
         <wptest.atech.com value=true />
      </Hosts>
```

Changing to 'true' will disable feature.

Add host or IP address to provide access.

Figure 8-1 Configuring for unknown host

These settings enable advanced caching as the default cache and allow connect tags to provide custom caching. Using connect tags to override the default cache is discussed in 3.7.2, "Configuring the <Cacher> configuration options" on page 117.

8.4 Configuring data caching

An under used feature of IBM Workplace Web Content Management is the ability to include external content into a rendered page via connect tags. External content can be from a HTML file, database results, LDAP entries, or DB2 Content Manager federated content. The external content is inserted just like a component, but it is not retained in the IBM Workplace Web Content Management repository. The data cache is a separate cache from the Web content cache, and it allows external content to be cached on the IBM Workplace Web Content Management server. The external content is requested from its remote location on the first request and then stored in the data cache. Subsequent requests for the external content are retrieved from the cache until it expires. Data cache must be configured separately from default cache.

You can configure the data cache in the connect.cfg file or in connect tag attributes. You can configure only the cache for HTML and database results in the connect.cfg. You must specify cache parameters for all other external content types in connect tag attributes.

To enable the data cache in the connect.cfg file, locate the `/Connector` section. The `/Connector/HttpConnector` section has three properties that control HTML external data cache:

- `/Connector/HttpConnector/DefaultCache`: If set to `true`, HTML data is stored in the site cache if no cache type is specified in the connect tag or request.
- `/Connector/HttpConnector/DefaultCacheExpires`: The default expiration time for HTML data if no expiration is specified in the connect tag or request.
- `/Connector/HttpConnector/OverrideCacheExpiryHeaders`: If set to `true`, the content expiration and cache setting in the HTTP headers is ignored.

Example 8-4 shows an example configuration.

Example 8-4 Sample HTTP Connector settings

```
<Connector>
...
   <HttpConnector>
   ...
      <DefaultCacheExpires value="REL 1D">
      <DefaultCache value=true >
      <OverrideCacheExpiryHeaders value=false >
   ...
```

The `/Connector/SQLConnector` section has one property which control the external data cache. The `/Connector/SQLConnector/DefaultCache` property determines if the site cache should be used if no cache parameter is supplied on the connect tag or request. Example 8-5 shows an example configuration.

Example 8-5 Sample SQL Connection settings

```
<Connector>
   ...
   <SQLConnector>
      ...
      <DefaultCache value=true >
      ...
```

You do not need to configure the data cache in the connect.cfg file, but it is recommended. Because each request for external data is unique and can have its own update frequency, personalization, and security needs, you should optimize each instance according to the data's requirements. By including cache type and expiration parameters on a connect tag, you can customize each data request. For more information about connect tags, see 8.6, "Custom caching with connect tags" on page 368 and the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

8.5 Configuring pre-rendering

Pre-rendering creates a snap-shot of the currently published site and can deliver the site via the IBM Workplace Web Content Management Servlet or an HTTP server. Pre-rendering is a batch process that generates the entire Web site each time it runs. Delivering a pre-rendered Web site via the IBM Workplace Web Content Management Servlet is enabled as the default module. To use an HTTP server, you need to configure pre-rending as a stand-alone module. (Refer to 3.7,

“Static pre-rendering” on page 114 for more information about how to configure static pre-rendering.)

Some important factors to consider before enabling pre-rendering are:

- If configured as the default module, the local rendering portlet cannot be used.
- Connect tags are not processed. Instead content is rendered to the file system as though the connect tags did not exist. If using the IBM Workplace Web Content Management local rendering server, connect tags are processed when the page is invoked.
- JSP components do not function. Pre-rendering does not invoke JSP components.
- Sites are rendered using the security context of the `RendererUser` set in the configuration. A RendererUser tells the `cacher` module to act as the specified user when processing content for pre-rendering. All content is pre-rendered as though the specified user was browsing the site. The user’s security permissions and personalization profile effects which content is pre-rendered.
- Only live content that is accessible by the RendererUser is pre-rendered.
- File system path and file name conventions must be adhered to in the names of Sites, Site Areas, and Pages. Only ASCI characters are permitted. You cannot use the following characters: **/ \ : * ?** " < > **|**
- File system path and file names cannot exceed the operating system’s maximum path length. For example, the Windows limit is 255 characters, while the Linux limit is 1024 characters.

These restrictions might not permit pre-rendering as viable caching solution for your Web site, but pre-rendering does offer extremely high performance gains. When configured as a stand-alone module, when the Web site is rendered to the file system, the HTML and resources can be distributed to other HTTP servers, removing even more load from IBM Workplace Web Content Management server.

8.5.1 Pre-rendering via an IBM Workplace Web Content Management local rendering server

The IBM Workplace Web Content Management local rendering server usually renders content as requested. When you enable pre-rendering, IBM Workplace Web Content Management saves the rendered content to the file system. The static content is then served from the local rendering server instead of building the content for each request.

To enable pre-rendering via the IBM Workplace Web Content Management local rendering server, you need to replace the default module's implementation class. In the connect.cfg file, locate the `/BusinessLogic/Module/Default` property and replace the class attribute value with `com.aptrix.cacher.CacherModule`, as shown in Example 8-6.

Example 8-6 Replacing the class attribute value with com.aptrix.cacher.CacherModule

```
<Default class=com.aptrix.cacher.CacherModule remoteAccess=true
autoLoad=false />
```

Now that the default module uses the `CacherModule` to serve content, the content can be accessed on the local machine using the following URL:

```
http://localhost:9081/wps/wcm/connect/<The Site Name>
```

Important: When you use the `CacherModule` as the default module, you cannot use the local rendering portlet to access the content.

8.5.2 Pre-rendering via an HTTP Server

You can configure IBM Workplace Web Content Management to pre-render the contents of an entire site to static HTML files. When pre-rendered, any Web server can deliver the content. Pre-rendering via an HTTP server requires that you configure a stand-alone module to render the static files and that you set up a Web server to deliver the content.

Pre-rendering via HTTP is a little more complicated than the IBM Workplace Web Content Management local rendering server. In the connect.cfg file, uncomment the `/BusinessLogic/Module/Cacher` property and uncomment the `/ModuleConfig/Cacher` property block. Next, configure the cacher module to render the site to disk. Table 8-4 list the properties and their descriptions.

Table 8-4 Property descriptions for pre-rendering via HTTP

Property	Description
`<DestDir value= "[ILWWCM_HOME]/ilwwcm/cacher" />`	The base directory under which each site cache is created. One subdirectory is created for each site.
`<TempDir value= "[ILWWCM_HOME]/ilwwcm/cacher/temp" />`	The temporary DestDir that is required to build the site cache prior to moving the data over to the DestDir.
`<Delay value="1" />`	This property is used to set the time, in seconds, between requesting a page while caching.

Property	Description
`<BusyDelay value="5" />`	This property is used to set the time, in seconds. It is used if executing within the BusyStart to BusyEnd period. Otherwise, the Delay setting is used.
`<BusyStart value="9:00 am" />` `<BusyEnd value="5:00 pm" />`	These settings determine the times between which the BusyDelay setting are used. Enter an absolute time.
`<Password value="password" />`	When manually pre-rendering a site via a URL request, you are asked for a password. This password is set here. It is recommended that you change the default password of *password* to something more secure.
`<OverWriteCache value="true" />`	If set to `true`, the pre-rendering overwrites files in the DestDir cache directory (then delete the unneeded files). This action results in a progressive change in site content that the user sees. If set to false, the pre-rendering renames the TempDir to the DestDir to achieve an instant update to a new version of the pre-rendered site. Do *not* use false should if you use an HTTP server to display the pre-rendered data because some HTTP Servers lock the data directories.
`<RendererUser value="Anonymous" />`	This determines the user to be used to render the Web Content Management content. Either type "anonymous" or "administrator" or a specific Member Manager user or group name. The site is pre-rendered based on this User's security rights. If the user specified here does not have access to a particular component it will not be pre-rendered.
`<CacherUrl value= "http://[HOST]:[PORT]/wps/wcm/connect/" />`	The full URL to the CacherModule.
`<ServletPath value="/connect" />`	The Servlet Path to the IBM Workplace Web Content Management servlet, not including the context path.
`<DefaultContentName value="index.html" />`	This property sets the name of the default or home file of the site that is used when accessing the pre-rendered site. This setting normally is index.html.

Archived

Property	Description
`<Sites value="Site1,Site2" />`	The sites within an IBM Workplace Web Content Management environment to cache are entered here, separated by commas.

`<Interval>`
`<Recurrence value="10" />`
`<StartDelay value="10" />`
`</Interval>`
	You can set the CacherModule to run after a recurring number of minutes.
	
	`Recurrence` is the recurring period in minutes for a recurring task.
	
	`StartDelay` is the delay in minutes prior to starting the first recurring task.

`<Scheduled>`
`<Times value="10:00am,12:00am,1:00pm" />`
`</Scheduled>`
	Alternately, you can set the CacherModule to run a certain times by entering a series of absolute times, separated by commas.

Note: If the pre-renderer is configured with `OverwriteCache` property set to `true`, all files located in the DestDir path that were not written by the pre-renderer are deleted. Therefore, only use the DestDir for storing rendered content.

In addition to configuring IBM Workplace Web Content Management for pre-rendering via HTTP, you must configure a Web server to serve the content. This Web server can be co-located with the IBM Workplace Web Content Management server, or it can exist on a remote server. IBM Workplace Web Content Management only renders content to a destination directory that is local to the IBM Workplace Web Content Management server. IBM Workplace Web Content Management cannot use other protocols (FTP, SCP, and so forth) to publish pre-rendered content. The destination directory can be a network mounted file system. Some operating systems, such as UNIX and Linux, support natively a wide variety of network file system protocols.

Regardless of where the Web server is located, IBM Workplace Web Content Management content contains path information that requires an alias entry in the Web server's configuration. The destination directory *must* be mapped to the alias wps/wcm/connect. This mapping ensures that links rendered in the IBM Workplace Web Content Management content are not broken.

8.6 Custom caching with connect tags

Connect tags are another function of the IBM Workplace Web Content Management engine that provide more control over the page rendering, caching, and dynamic content federation. This section discusses how you can use connect tags to implement custom caching. Connect tags do provide other features. However, those feature are beyond the focus of this section. To learn more about other features that connect tags provide, see the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

Connect tag syntax is a replacement to IBM Workplace Web Content Management component tags (`AptrixCmpnt` and `AptrixLibCmpnt`) and provides attributes to override the site's default cache type and expiration settings. The connect tag syntax is as follows:

```
<connect mod="" srv="" {<Additional attributes>=""}>{Failure
text}</connect>
```

Table 8-5 lists the available attributes and options.

Table 8-5 Attributes and options for connect tags

Attribute	Description
`mod`	Module used to display content. There are several valid modules, but for caching content in IBM Workplace Web Content Management, only the Default and ComponentRender options are needed. If this attribute is omitted, the Default module is used.
`srv`	Indicates the data provider for displaying content. The `cmpt` option selects content or library components.
`path`	If retrieving a menu or navigator, this attribute supplies a path context (for example, /Site/Site Area/Content).
`source`	Specifics from where to retrieve the component. Valid options are: `Library`, `Site`, `Site Area`, and `Content`.

Archived

Attribute	Description
`cache / contentcache`	Overrides the site's default cache type, and indicates the cache type to use. Valid options are `Site`, `Session`, and `None`. These options are only valid when advance caching is used with the following options: `User`, `Secured`, and `Personalized`. Use the `cache` attribute with basic default cache. Use the `contentcache` attribute with advance or disabled default cache.
`expires / contentcacheexpires`	Overrides the site's default cache expiration and indicates a new expiration date or time period. Uses the expiration syntax as described in 3.7.2, “Configuring the <Cacher> configuration options” on page 117. Uses the `expires` attribute with basic default cache. Uses the `contentcacheexpires` attribute with advance or disabled default cache.
`cmpntname`	Name of component to retrieve content from

You can use connect tags as a replacement for the IBM Workplace Web Content Management component tags. The following examples replace the IBM Workplace Web Content Management component tags with connect tags and add caching attributes.

Example 1: Replacing the library HTML component

This sample uses the component tag:

```
<AptrixLibCmpnt name="HTML - Home Page Navigation" />
```

This sample uses connect tags and adds caching attributes:

```
<connect mod="ComponentRenderer" srv="cmpnt" source="library"
cmpntname="HTML - Home Page Navigation" contentcache="site"
contentcacheexpires="REL 9000s" ></connect>
```

Example 2: Replacing the content component

This sample uses the component tag:

```
<AptrixCmpnt context="current" type="content" key="Related Pages" />
```

This sample uses connect tags and adds caching attributes:

```
<connect mod="ComponentRenderer" srv="cmpnt" source="content"
cmpntname="Related Pages" contentcache="personalized"
contentcacheexpires="REL 2H" ></connect>
```

Example 3: Replacing the component reference in matching content of navigator or menu component

This sample uses the component tag:

```
<AptrixCmpnt context="autoFill" type="content" key="Display Title" />
```

This sample uses connect tags and adds caching attributes:

```
<connect mod="ComponentRenderer" srv="cmpnt" source="content"
path="<placeholder tag='sitepath' />/<placeholder tag='name' />"
cmpntname="Display Title" contentcache="secured" contentcacheexpires="REL
1D" ></connect>
```

Note: A connect tag does not have an *autoFill* option. To set the current path matched by a navigator or menu component, a connect tag *must* use placeholder tags in its path attributes.

Connect tags are the center piece of a custom caching strategy. These tags offer tremendous flexibility and performance improvements. However, they require discipline and practice. It is tempting to replace IBM Workplace Web Content Management component tags with connect tags throughout the site, without analyzing the site's performance first. Remember, connect tags are not processed in some circumstances. Be sure you use connect tags where it makes sense. When first using IBM Workplace Web Content Management, figuring out where to use connect tags should be driven by where the site's performance merits their use. Conduct a performance baseline, and then consider using connect tags to override the site's default cache.

Archived

8.7 Considerations in developing a caching strategy

The previous sections defined IBM Workplace Web Content Management caching capabilities and demonstrated how to configure and to use caching. This section provides tips and recommendations to help you develop a caching strategy. It includes the following sections:

- Identifying performance impacts
- Cache recommendations for components types
- Other caching systems that can effect performance

8.7.1 Identifying performance impacts

Presentation templates and components offer a great degree re-use. However, the number and complexity of the components that are included on presentation templates can affect the performance of the IBM Workplace Web Content Management page rendering process. Often, the same components are processed and inserted into multiple pages repeatedly. During the development of a caching strategy, it is important to understand the performance effects from component aggregation.

Identifying which components affect performance begins with the component or tag's function. The most compute intensive functions in IBM Workplace Web Content Management rendering are aggregation dynamic content inclusion and external content inclusion. Your caching strategy needs to consider the impact of the function before determining if and how an element should be cached.

Aggregation is the referencing of one IBM Workplace Web Content Management object within another. The process begins with the presentation templates that include components, fields, metadata, and connect tags. The aggregation continues as components include other components — generally called *composite* components. The composition depth can reach many levels deep, depending on the complexity of the component. A good cache strategy practice is to cache the composite component at the highest level possible (that is, closer to the reference in the presentation template is better).

In addition, you should consider how a composite component is re-used by other composite components. A composite component's security and personalization requirements also effect the decision of where to implement caching. If caching is implemented too low in the composition depth the cache might not save processing or memory, and if implemented too high in the composition depth, the cache can prevent dynamic content from updating appropriately.

Archived

Because composite components can include just about any other component or connect tag, these components can implement any of the caching types. Clearly defining the functionality for each element helps when the time comes to decide how to cache the element.

The dynamic functionality of IBM Workplace Web Content Management depends on the run-time state to determine what content to render. Evaluating dynamic functionality, achieved through components and connect tags, is one of the most compute expensive operations in IBM Workplace Web Content Management. Implementing caching on dynamic functionality improves performance significantly. The primary consideration for dynamic functionality is how dynamic must it be?

The update frequency of a dynamic element is defined in the site's information architecture. Often, the architecture defines how frequently a news feed should be refreshed for new content. However, elements such as the left-hand navigation or related links list are not defined. Because these elements are commonly represented as navigator and menu components, respectively, it benefits performance if these components are cached.

Dynamic functionality usually evaluates group memberships, category selections, keywords, or a personalization profile, and then delivers customized content. Selecting which cache type to use is a matter of the functionality that is required. For example, if the content needs to respond to the user's group memberships, then use the secured cache type. Dynamic functionality usually implements session, secured, user, or personalized cache types.

Repurposing external content in IBM Workplace Web Content Management by referencing it with a component or connect tag is an expensive operation. IBM Workplace Web Content Management retrieves content from a remote system and formats it for display within a presentation template. Functionality remains the primary concern. However, because a remote system is the source, a caching strategy must also consider the effect of the remote system that is temporarily unavailable. In this case, cache can mitigate the impact of temporary outages.

In addition, formatting the remote content can be expensive. It might require multiple requests to the remote system if only one attribute is retrieved at a time. When dealing with external content, it is preferable to format it and then cache the result when using a site cache type. Of course, if the external content responds to the current user's run-time state, then a more dynamic cache type is necessary.

8.7.2 Cache recommendations for components types

The following list includes our recommendations for implementing cache on some content and library component types:

HTML components

These components make excellent root containers for composite components. In addition to HTML, these components can contain IBM Workplace Web Content Management component tags and connect tags. As a result, HTML components are useful in aggregating other components via IBM Workplace Web Content Management component tags. It is a good practice to cache these composite components and to reference them with a connect tag, as shown in the following examples:

- HTML component: HTML — Home Page Navigation

```
<div>
<AptrixLibCmpnt name="NAV - Products" />
<AptrixLibCmpnt name="NAV - Services" />
<AptrixLibCmpnt name="NAV - Support" />
</div>
```

- Presentation Template — Home Page Layout (using basic caching)

```
<html>
…
<body>
<connect srv="cmpnt" source="library" cmpntname="HTML - Home Page
Navigation" cache="site" expires="REL 12H" >
</connect>
```

If a composite component needs different caching requirements on the components that it aggregates, then HTML and Rich Text components offer flexibility. Each connect tag can use different caching types and expiration parameters, as shown in the following examples:

- HTML — HTML, right-hand navigators:

```
<div>
<connect srv="cmpnt" source="library" cmpntname="MENU - Hot Picks"
cache="session" expires="REL 1H" ></connect>
<connect srv="cmpnt" source="library" cmpntname="NAV - Site Tools"
cache="site" expires="REL 24H" ></connect>
</div>
```

- HTML component— HTML, right-hand navigation

```
<div>
<connect srv="cmpnt" source="library" cmpntname="MENU - Hot Picks"
cache="session" expires="REL 1H" ></connect>
```

Archived

```
<connect srv="cmpnt" source="library" cmpntname="NAV - Site Tools"
cache="site" expires="REL 24H" ></connect>
</div>
```

Text components

Text components can contain static HTML and connect tags. If a connect tag is re-used throughout the site, instead of pasting the tag for each instance, the connect tag and its parameters can be saved in a text component and referenced using IBM Workplace Web Content Management component tags. This provide re-use, but if a parameter is changed, all occurrences are updated.

JSP components

Because JSP components can implement such a wide range of functionality, selecting a cache type is not always a clear choice. JSP components tend to deliver either custom formatting logic or integration with an external system. Determining a cache type usually depends on the level of dynamic behavior and whether the content is personalized.

Note: A JSP component is not rendered if a pre-rendering strategy is implemented.

Menu and navigator components

These two components are performance intensive functions of IBM Workplace Web Content Management. Menus tend to be far more performance-intensive than navigators, because of the nature of their more flexible content selection possibilities. How the site's menu and navigator components behave often is the primary factor for deciding between basic and advance caching. If these components are required to respond to the current user's group memberships or profiling attributes, then an advanced cache strategy might be necessary. Typically, you can implement a site cache strategy and then override the cache for menus and navigators.

Personalization components

These components are new to IBM Workplace Web Content Management 5.1.0.1, and their behavior is similar to personalized menu components. Personalization components use personalization rules and content spots that are defined in WebSphere Portal's personalization engine, to aggregate and to deliver IBM Workplace Web Content Management and WebSphere Portal content.

Caching personalization components override the default cache and implement personalized cache type on a individual component basis. Unlike menus, which use IBM Workplace Web Content Management content or content that is

federated into IBM Workplace Web Content Management, personalization components display content that is aggregated in a portal personalization rule or content spot. Personalization rules and content spots can use IBM Workplace Web Content Management content but can also include external content.

Note: If connect tags are used within HTML, text, and Rich Text components, ensure that Process Connect Tags is selected on the presentation template that includes these components.

8.8 Other caching systems that can effect performance

Before implementing or testing a caching strategy, it is important to be aware of other caching systems which might effect IBM Workplace Web Content Management performance. If IBM Workplace Web Content Management is delivering content through WebSphere Portal, consult the portal administrator and determine if portal pages that deliver IBM Workplace Web Content Management content are configured to be cached by the Portal.

Note: The Portal's caching system is beyond to scope of this material. To learn more about Portal's cache system, consult the WebSphere Portal Information Center at:

http://www-306.ibm.com/software/genservers/portal/library/index.html

In addition to Portal's cache, review the cache settings on any Web servers that deliver IBM Workplace Web Content Management content that is configured with the WebSphere Application Server Plug-in. Basic caching should not be effected by the Web server's cache. However, if you are using an advanced cache strategy, ensure that the Web server's cache settings are configured to deliver secure and personalized content. Otherwise, performance can degrade because your Web server is caching too much or is rebuilding the cache constantly.

Be aware of how each system cache's content. Otherwise, unexpected results can appear when implementing and testing your cache strategy.

A caching strategy alone does not always improve performance sufficiently. We recommend that you follow WebSphere Application Server and WebSphere Portal Server performance tuning procedures in addition to implementing a caching strategy.

8.9 Documenting the cache strategy

Developing a cache strategy requires gathering information about the usability requirements and IBM Workplace Web Content Management implementation. Distilling all this information into a format to assist with site analysis can be a difficult task to manage. IBM Workplace Web Content Management does not provide an interface for tracking requirements and functionality. Producing cache strategy documentation is just another task on par with developing wire-frames, metadata schemas, or taxonomies.

Ideally, the cache strategy documentation should be included with the information architecture. The architecture should describe the functions of elements on the screen and include descriptions about how a user interacts with the elements. These elements are translated into templates and components in IBM Workplace Web Content Management.

After elements are realized as objects in IBM Workplace Web Content Management, the first step is to begin a catalog of components and presentation templates. The catalog can be simple spreadsheet workbook. Next, create a caching events timeline for each presentation template. You can create the events timeline using any flow chart application. If the site's architecture requires customized content to a session, user group, or personalization profile, it might be help to create diagrams with a timeline for each scenario. The following sections provide explanations of the documentation and examples of the information that you need to collect.

Before continuing, the volume and complexity of your documentation should reflect the size and complexity of the site. Some of the recommendations made in this section assume that the site in IBM Workplace Web Content Management is of sufficient size and complexity to warrant detailed documentation. Small sites might not require this level of documentation. Also, the cache type and expiration interval values use the same syntax as the values that are used in the connect tag cache and expires attribute.

8.9.1 Component and template catalog

The component and template catalog provides a list of all components in IBM Workplace Web Content Management and the presentation templates that use these components. Developing a catalog and updating it regularly is no small task. However, having a single reference to how the site's caching strategy is implemented is an invaluable resource. You can enter the list of library components and presentation templates manually, or you can export them programmatically via the IBM Workplace Web Content Management API.

Presentation template entries should include a unique entry ID, name, and description. Some of this data is stored in IBM Workplace Web Content Management as metadata to each presentation template. The unique entry ID is an ID that is unique for the catalog, as described in Table 8-6.

Table 8-6 Unique entry ID

Entry ID	Name	Description
T-01	Details Page Layout	HTML presentation template for Details authoring template. Default template for site.
T-02	Menu Page Layout	HTML presentation template for Menu authoring template. Defines Layout for Landing pages.

Each component entry in the catalog should include a unique entry ID, name, description, cache type, expiration interval, and a list of presentation template entry IDs. Some of this data is stored in IBM Workplace Web Content Management as metadata to each component. The cache type and expiration interval most likely is empty when you build the catalog. Make entries into the cache type and expiration when the site's default cache is overridden. Table 8-7 shows an example of a component entry.

Table 8-7 A component entry

Entry ID	Name	Description	Cache Type	Expiration Interval	Used in
C-01	NAV - Products	Displays all child site areas of the Products Site Area			T-02
C-02	Nav - Left Hand	Displays all child site areas of the Services site area	Site	REL 24H	T-02, T-03

Archived

Cache events timeline

After collecting the data for the component and template catalog, the information can be represented graphically on a timeline. Creating a visual timeline is another aid to understanding how the site's cache strategy functions. The data can be visualized in whatever format desired. Figure 8-2 uses a simple bar graph.

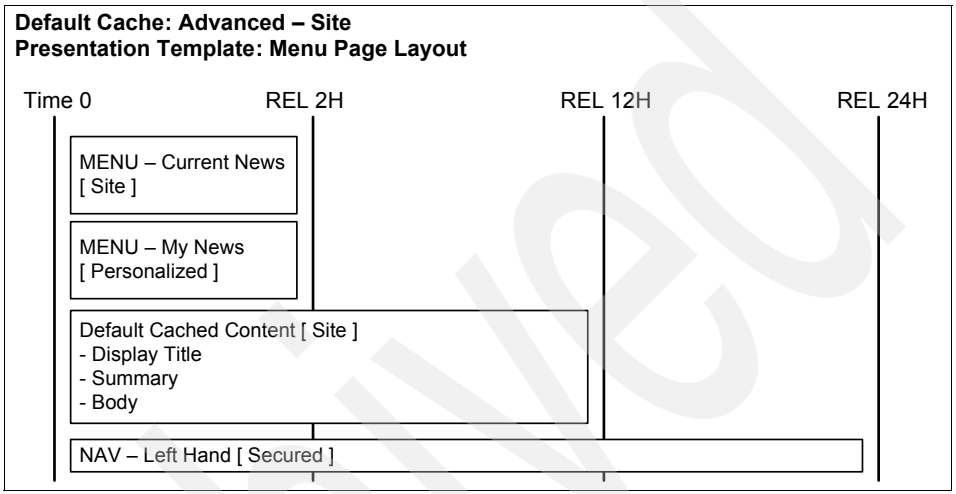


Figure 8-2 Example of a visual timeline in a bar-graph format

Figure 8-2 illustrates the cache refresh timeline for content that is rendered using the presentation template Menu Page Layout. The diagram defines:

- Time zero, which signifies the first time that the page is rendered. The page and components are stored in cache.
- Every 2 hours, the components `MENU- Current News` and `MENU - My News` expire from cache. The next time either of the components is rendered, it is cached. `MENU - Current News` implements site cache type, and `MENU - My News` implements personalized cache type.
- Every 12 hours, content that uses the default cache expires. The next time the page is accessed, it is cached.
- Every 24 hours, the component `NAV - Left Hand` expires from cache. The next time the component is rendered, it is cached. The component overrides the default cache and implements secured cache type.

Figure 8-2 depicts the case timeline for only one presentation template. You can expand this diagram easily to include multiple presentation templates and the cache dependences between them, as illustrated in Figure 8-3 on page 379.

Archived

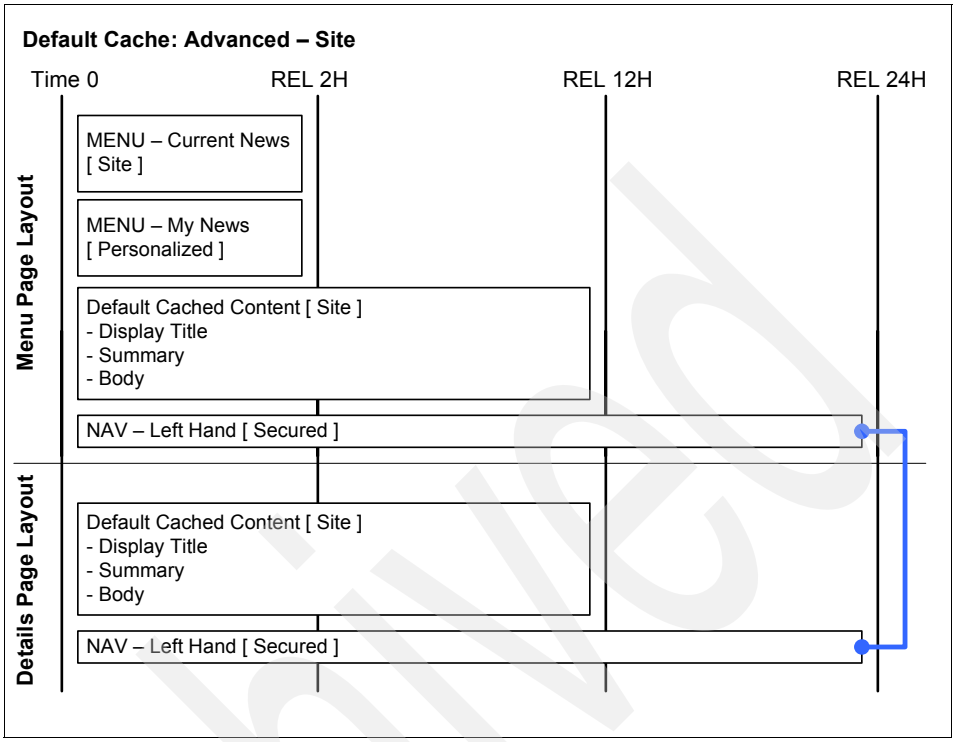


Figure 8-3 Two presentation templates and their cache dependences

Figure 8-3 expands the previous figure and includes a second presentation template. The cache refresh timeline is the same. However, this figure illustrates how the `NAV - Left Hand` component shares the same cache in both presentation templates.

> **Note:** These diagrams only displayed relative times. You can also set the IBM Workplace Web Content Management cache to absolute times.

Time invested in developing a complete information architecture and supporting documentation pays off when it comes time to manage and to update the site. The cache strategy is just one aspect of the architecture to document. Creating visual representations of the caching strategy can communicate functionality to stakeholders and other interested parties who are not familiar with IBM Workplace Web Content Management.

Archived

8.10 WebSphere Dynamic Cache Service

Dynamic Cache (WebSphere Dynamic Cache Service) is a WebSphere service that is enabled on an application server by default. It supports caching of Servlet and JSP responses, WebSphere Commands objects, Web services objects, and Java objects. As an enterprise application on the WebSphere Application Server, the IBM Workplace Web Content Management uses this service.

To improve performance for an IBM Workplace Web Content Management site, there are a number of options available from the internal caching and pre-rendering to servlet caching within IBM Workplace Web Content Management. While this section provides a quick review of potential caching options, including IBM Workplace Web Content Management pre-rendering and IBM Workplace Web Content Management caching, it provides an overview of the WebSphere Dynamic Cache Service. It also provides a step-by-step guide to setting up WebSphere Dynamic Cache to work with IBM Workplace Web Content Management sites (portal and non-portal).

8.10.1 High performance (caching) options

This section serves as a brief review of the caching options and describes each option. It also introduces the WebSphere Dynamic Cache Service.

IBM Workplace Web Content Management pre-rendering

Consider the following before implementing an IBM Workplace Web Content Management pre-rendering strategy:

- Pre-rendering refers to a function within the IBM Workplace Web Content Management application that collates the design and content components and then stores them on a physical disk as distinct and complete HTML files.
- This option provides extremely fast response because the HTML files are placed as close as possible to the site visitors. For a portal site, use this option with Web clipping or other HTML portlets but not with the IBM Workplace Web Content Management rendering portlets.
- The IBM Workplace Web Content Management engine renders an entire site (based on the site framework that is selected in the configuration file) and cannot be broken down into partial pre-rendering. Pre-rendering is done by enabling the `<Cacher class=….>` module in the connect.cfg file. The resulting HTML files can be stored on another device and used by other Web servers.
- This option cannot provide caching for dynamic caching (including IBM Workplace Web Content Management JSP component), personalized, or secured content.

For more information about this strategy, see 8.1.2, “Pre-rendering options” on page 351.

Figure 8-4 illustrates this strategy.

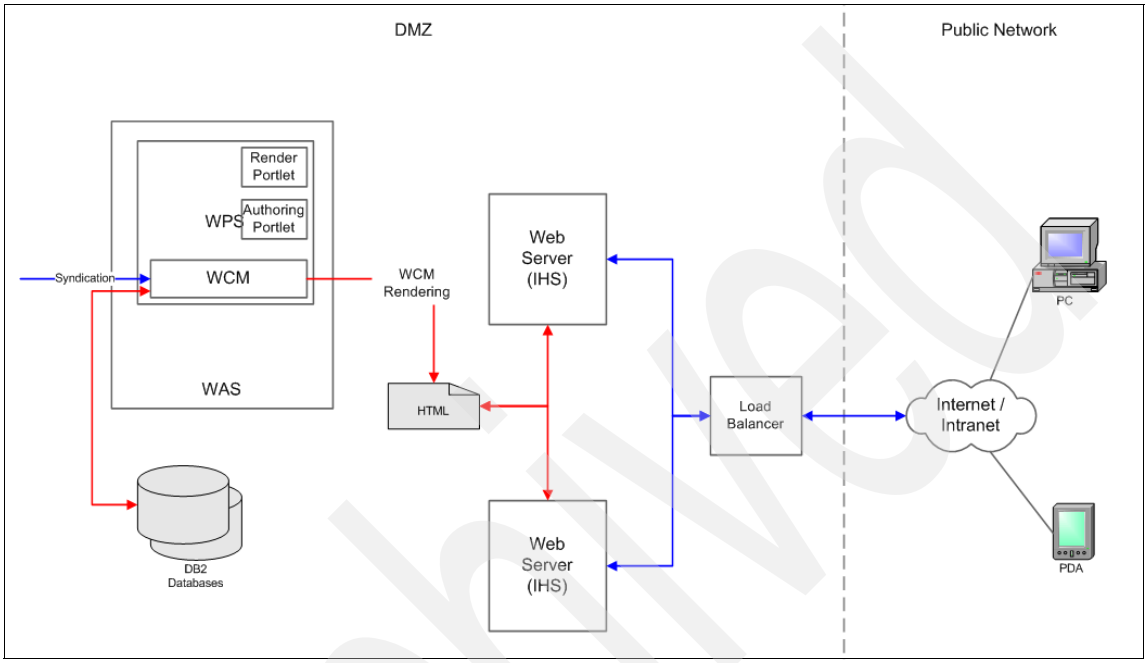


Figure 8-4 IBM Workplace Web Content Management pre-rendering strategy

IBM Workplace Web Content Management caching

Consider the following before implementing an IBM Workplace Web Content Management caching strategy:

- IBM Workplace Web Content Management caching stores content (after it has been rendered from various IBM Workplace Web Content Management components). Cache is stored when a page is requested.
- IBM Workplace Web Content Management Caching is set up through the cache configuration in the connect.cfg file. It requires IBM Workplace Web Content Management to act as a Web server because the caching file system in not in a true HTML format.
- Basic caching is similar to pre-rendering but offers a more flexible solution for a larger Web site with better currency of content. However, it cannot provide cache for dynamic, personalized or secured content.
- This option is also fairly fast but can only be used for non-portal sites.

Archived

- Advanced caching, which provides options to cache content selectively by session, users, and so forth, provides same output as the basic caching option.
- In addition, you can use connect tags to cache (or un-cache) components selectively within a Web page. This option is useful for a Web site that has content that is largely static with some dynamic components.

For more information about this strategy, see 8.1.1, “Caching options” on page 349.

Figure 8-5 illustrates this strategy.

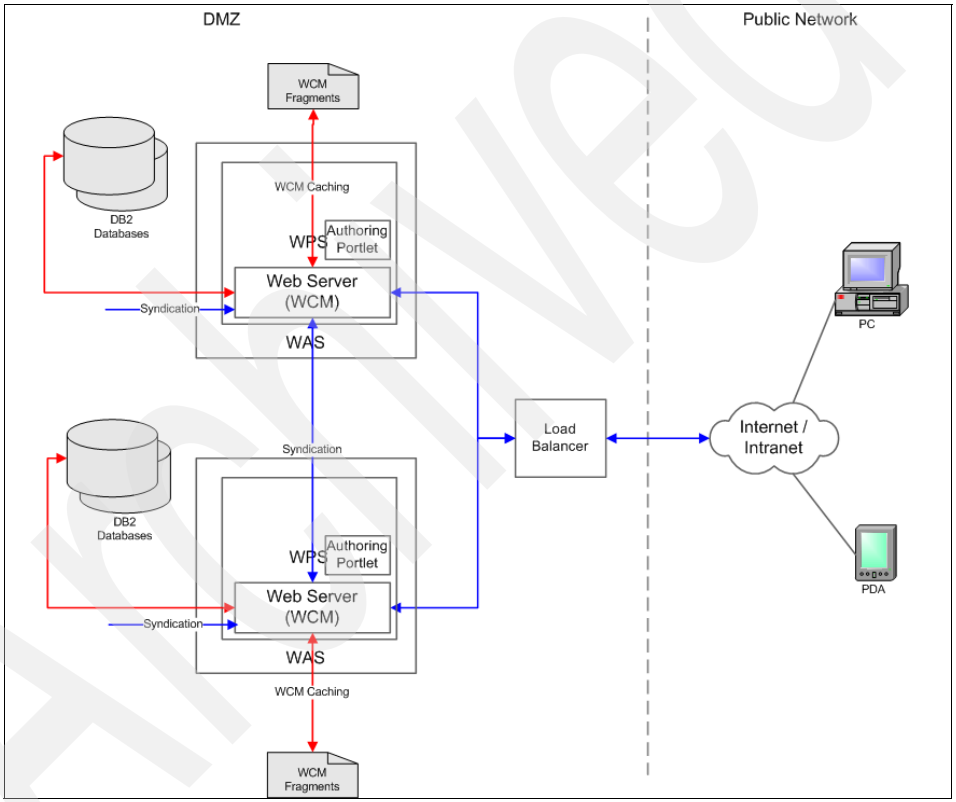


Figure 8-5 IBM Workplace Web Content Management caching strategy

Archived

Dynamic cache with servlet caching

Consider the following before implementing dynamic cache with servlet caching:

- This option represents an IBM Workplace Web Content Management caching solution that uses WebSphere Dynamic Cache Service.
- Servlet caching must be enabled on the WebSphere Application Server.
- For WebSphere Application Server V5, only one cache policy file is allowed per IBM Workplace Web Content Management instance. For WebSphere Application Server V6, it is possible to create multiple *servlet cache instances* to provide better flexibility and tuning of cache resources.
- This option offers a flexible and configurable solution based on WebSphere Application Server caching. You can configure a number of policy values to provide a tailored caching option, such as site structure, expiration, exclusion, and so forth.
- This option can cache static or dynamic content as fragments. Fragments can be set up to expire at different periods or to not cache at all.
- You can use this option for both portal (rendering portlets) and non-portal sites.
- This option can be configured to offload cache to disk if the memory cache is full (Disk Offload function).
- There is a limited cache memory management.
- In addition to the servlet caching, this caching solution can be further extended by using the Caching Proxy (part of Edge Components).

Archived

Figure 8-6 illustrates this caching strategy.

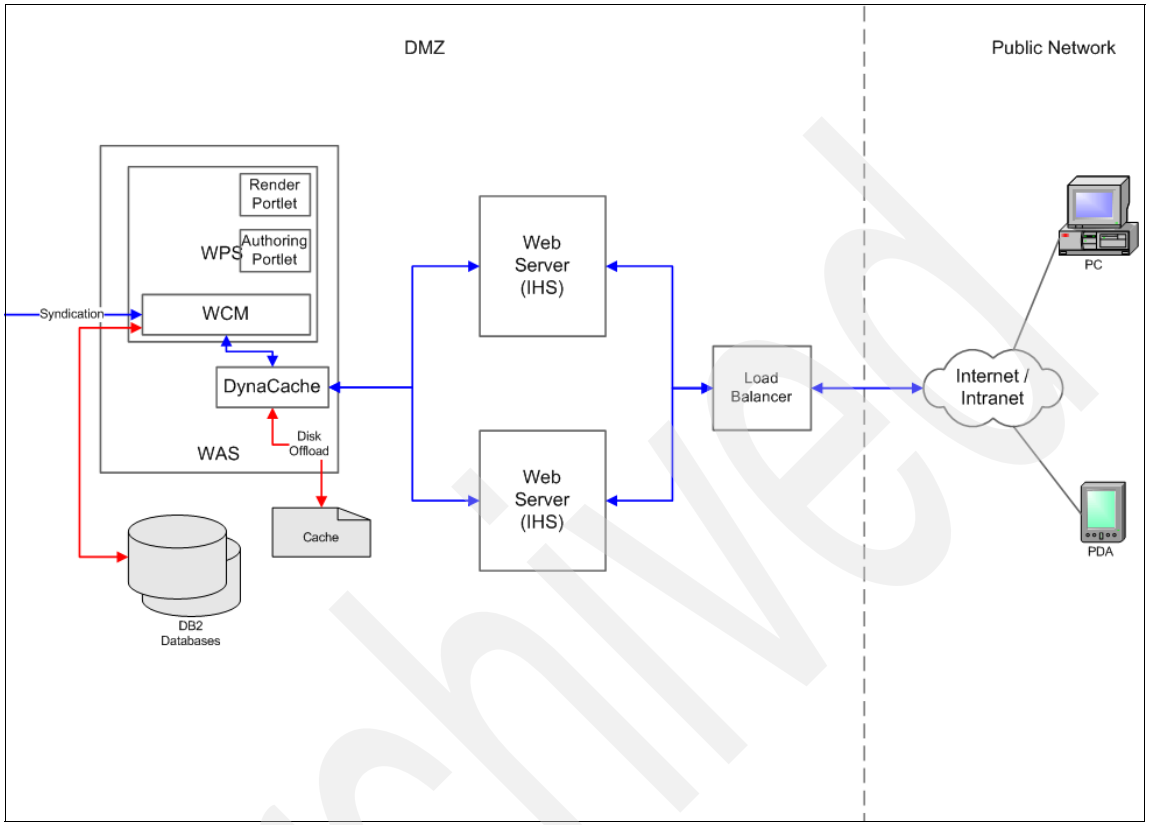


Figure 8-6 Servlet caching (dynamic cache)

Edge components

Consider the following before implementing edge components in your caching strategy:

- This option is an extension to the servlet caching.
- The caching proxy provides a caching solution, but is placed typically in front of the Web servers (and closer to the users).
- There are plug-ins to integrate with the directory to provide authentication and authorization.
- This method can also be cached at the edge (Web server level) if a plug-in is enabled. Invalidation options include timeouts, invalidation policy, or explicitly through the Edge Side Include API.

Archived

Figure 8-7 illustrates this caching strategy.

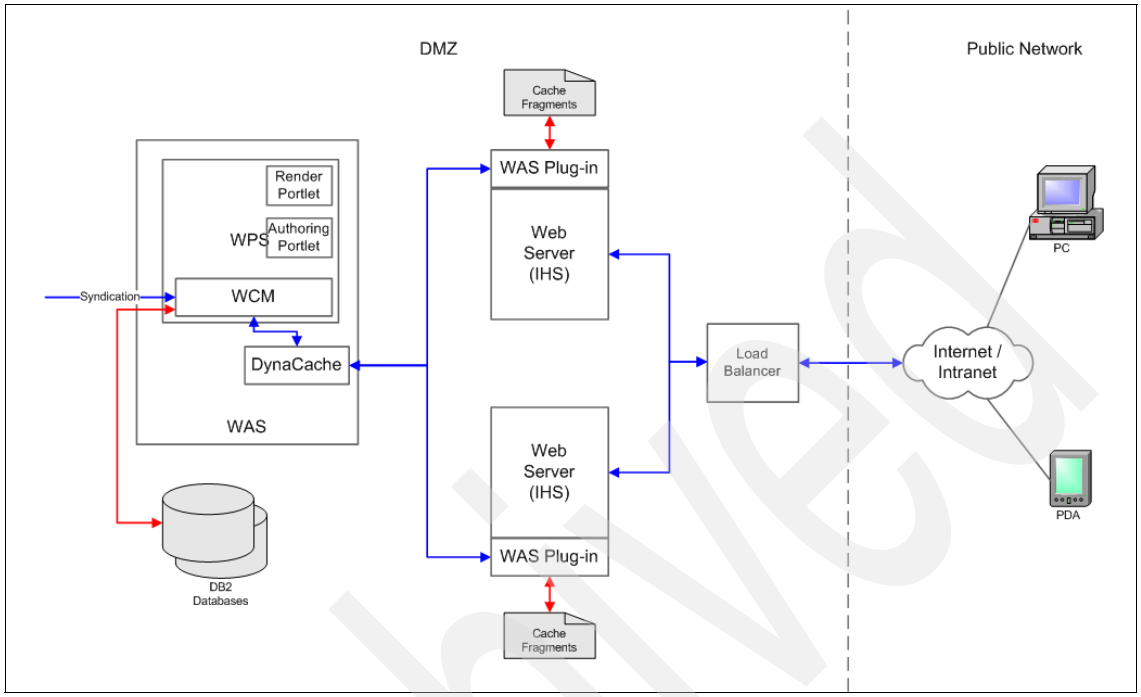


Figure 8-7 Dynamic cache with (Edge Side Include)

Dynamic cache with DistributedMap

DistributedMap is an interface that is provided by dynamic cache that applications use to cache and to share Java objects. This option is added for completeness, and we do not discuss it here. For more information about the DistributedMap option, refer to the WebSphere Application Server Information Center at:

http://publib.boulder.ibm.com/infocenter/wasinfo

8.10.2 Dynamic cache install and configuration

This section describes how to install and configure Dynamic Caching for WebSphere Portal. In addition to the description provided here, we recommend that you also refer to the WebSphere Portal Information Center at the following Web address for the latest information:

http://www-306.ibm.com/software/genservers/portal/library/index.html

Enabling servlet caching on a portal server

You must enable servlet caching on a portal server because IBM Workplace Web Content Management is a servlet by following these steps:

1. Go to the WebSphere Administration Console and select **Servers** → **Application Servers** → **WebSphere Portal**.
2. Select **Enable servlet caching** as shown in Figure 8-8.

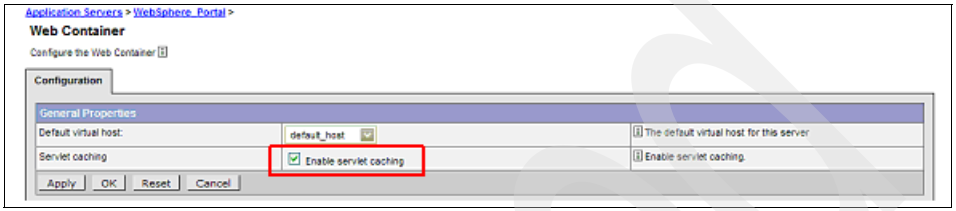


Figure 8-8 Enabling servlet caching

Note: This action requires that you restart the server. However, you should wait to restart the server until you have made all changes to the configuration.

Installing the dynamic cache monitor

The dynamic cache monitor is an installable Web application that displays simple cache statistics, cache entries, and cache policy information for servlet cache instances. The cache monitor is not installed on the WebSphere Application Server by default. To install it manually, follow these steps:

1. Locate the CacheMonitor.ear file (Figure 8-9). This file is usually located in the installedableApps directory of the [*WAS HOME*] directory, for example C:\Program Files\WebSphere\AppServer\installableApps.

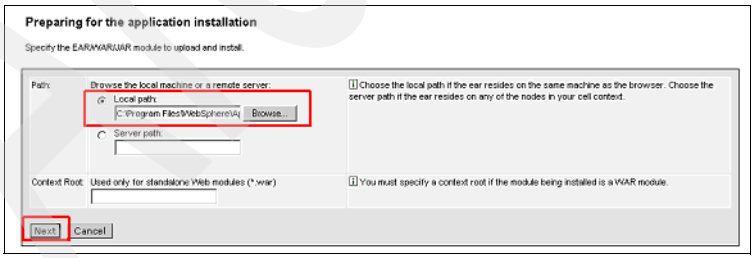


Figure 8-9 Locating the CacheMonitor.ear file

2. From the main WebSphere Application Server Administration Console, select, **Applications** → **Install New Application**.

3. Choose Local Path and use the Browse button to locate the Cache Monitor install file. Click **Next** to proceed.
4. Accept the default settings, as shown in Figure 8-10.

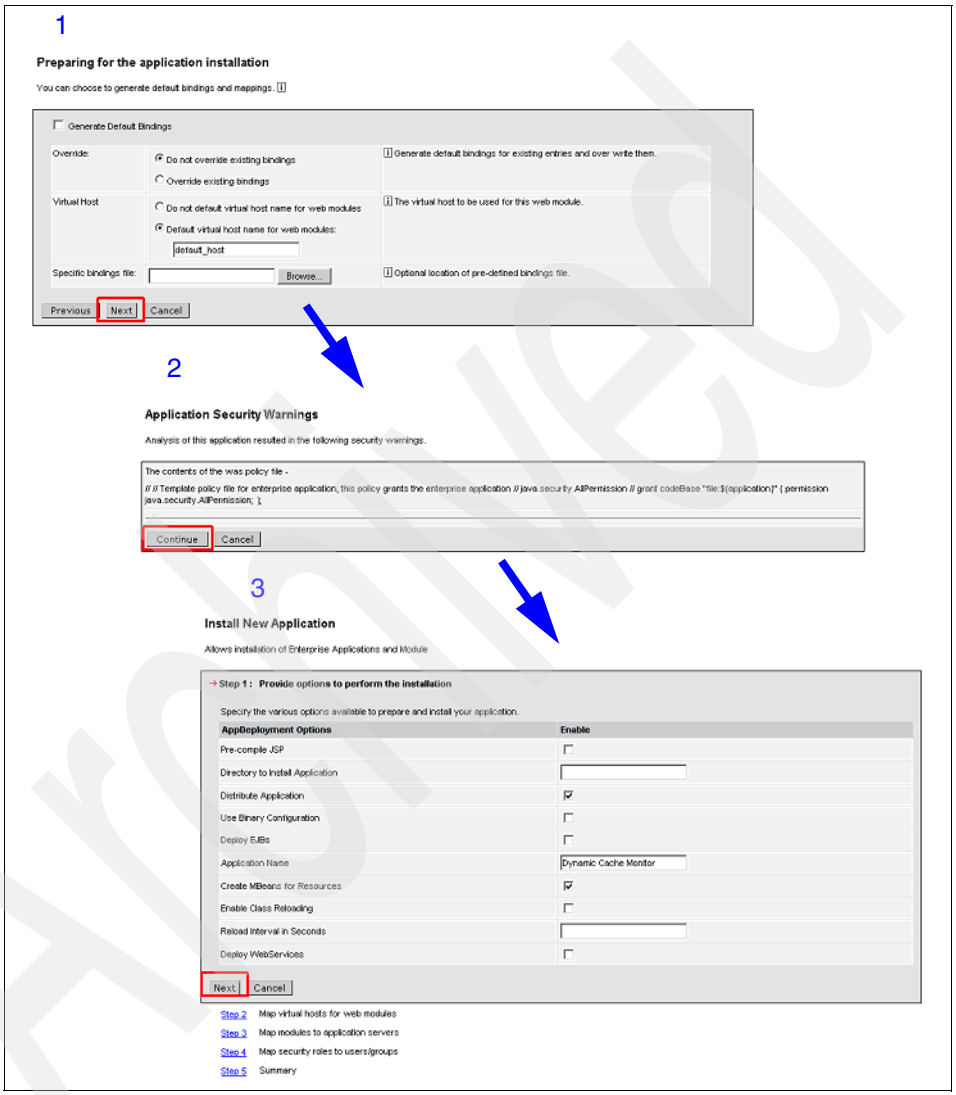


Figure 8-10 Default settings for installation screens

5. Map the cache monitor to a virtual host. Figure 8-11 on page 388 uses the `default_host`. (You can use the portal host name and port to access the cache monitor application.)

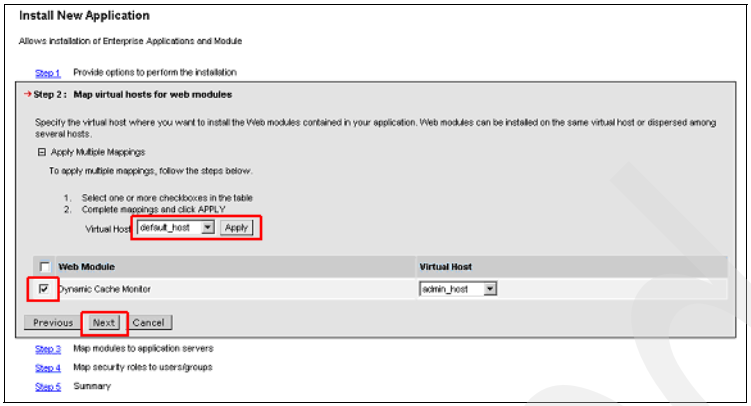


Figure 8-11 Mapping the cache monitor to a virtual host

6. Map the Web modules to the servers to be monitored (for example, WebSphere_Portal and server1) as shown in Figure 8-12.

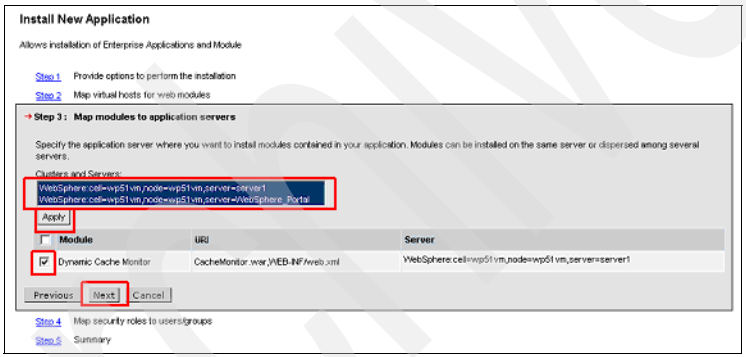


Figure 8-12 Mapping Web modules to the servers to be monitored

Archived

7. Set up security access to the Interface. In Figure 8-13, access is given to everyone. However, in the production environment, you might want to restrict this access to the Administrator group.

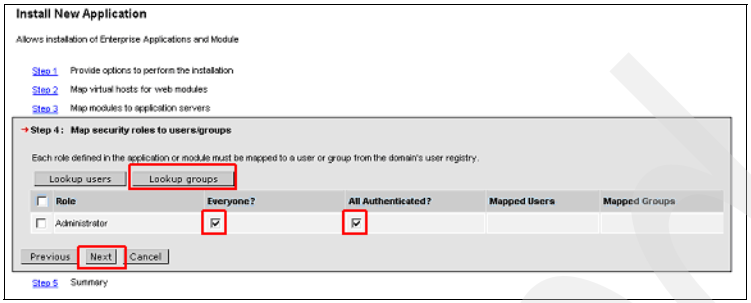


Figure 8-13 Setting up security access to the interface

8. Click **Finish**, **Save to Master Configuration**, and **Save** again, as shown in Figure 8-14 on page 390.

Archived

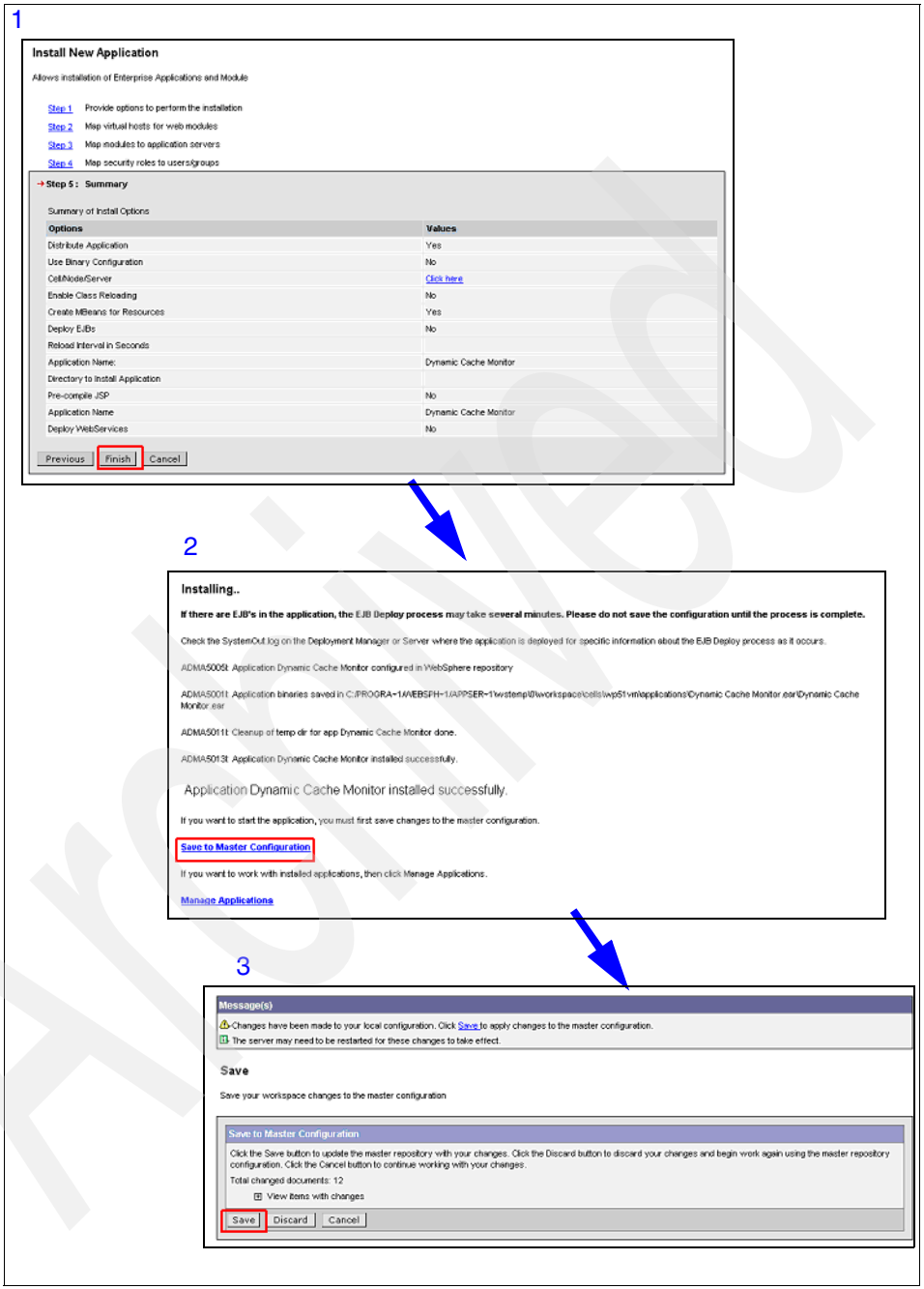


Figure 8-14 Completing installation and saving to master configuration

Archived

Configuring IBM Workplace Web Content Management cache policy

The cache policy is stored as an XML file (cachespec.xml) on the IBM Workplace Web Content Management installed sub directory, for example <*WAS_ROOT*>\installedApps\<server_name>\ilwwcm_war.ear\ilwwcm.ear\WEB_INF\. Example 8-7 caches all content except for Search Results, Personalize, and Voice Mail Policy pages.

Example 8-7 Cache policy example

```
<?xml version="1.0" ?>
<!DOCTYPE cache SYSTEM "cachespec.dtd">
<cache>
   <cache-entry>
      <class>servlet</class>
      <name>com.presence.connect.ConnectServlet.class</name>
      <property name="store-cookies">false</property>
      <property name="save-attributes">true</property>
      <property name="consume-subfragments">true</property>
      <cache-id>
         <component id="" type="pathinfo">
            <required>true</required>
            <not-value>/Intranet/Search+Results</not-value>
            <not-value>/Intranet/Tools/Personalize/</not-value>

<not-value>/Intranet/Home/IT/Procedures/Voice+Mail+Policy</not-value>
         </component>
         <timeout>3000</timeout>
      </cache-id>
   </cache-entry>
</cache>
```

In this example:

- The `com.presence.connect.ConnectServlet.class = container` class for IBM Workplace Web Content Management caches everything except those pages that have been excluded explicitly.
- The exclusion value (in `<not-value>`) must contain the forward slash (/) at the beginning of the policy value. Otherwise, there is unpredictable behavior for the home page (`error 500 - Server caught unhandled exception from servlet [WCM Framework]: OutputStream already obtained`).
- A forward slash (/) at the end of a policy value to specify whether the children of the path is included on just the components at that level. For example:

  ```
  <not-value>/Intranet/Tools/Personalize/</not-value>
  ```

Archived

DynaCache interprets this as any objects and components with a URI that starts with /Intranet/Tools/Personalize/, such as the following:

```
<not-value>/Intranet/Home/IT/Procedures/Voice+Mail+Policy</not-value>
```

This example exclude only the Voice Mail Policy page from the cache.

Alternatively, you can use the policy in Example 8-8. This policy is particularly useful if the caching solution is required only for part of the site (or other site framework).

Example 8-8 Caching for only part of the site

```
<?xml version="1.0" ?>
<!DOCTYPE cache SYSTEM "cachespec.dtd">
<cache>
   <cache-entry>
      <class>servlet</class>
      <name>/connect/Intranet/Home/</name>
      <property name="store-cookies">false</property>
      <property name="save-attributes">true</property>
      <property name="consume-subfragments">true</property>
      <cache-id>
         <component id="" type="pathinfo">
            <required>true</required>
            <not-value>/Intranet/Search+Results</not-value>
            <not-value>/Intranet/Tools/Personalize/</not-value>

<not-value>/Intranet/Home/IT/Procedures/Voice+Mail+Policy</not-value>
         </component>
         <timeout>3000</timeout>
      </cache-id>
   </cache-entry>
</cache>
```

In this example, The value for `<name>` must be in the following format:

```
/connect/[SITE]/[HOMEPAGE]/
```

Archived

Figure 8-15 illustrates an example site framework and its site areas.

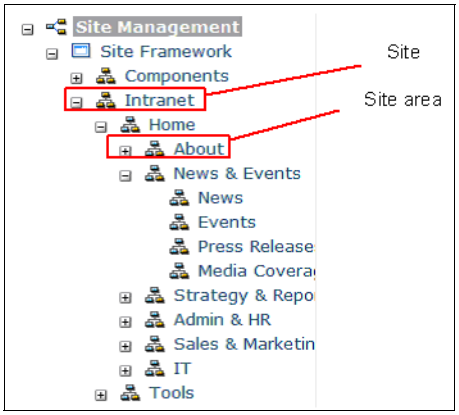


Figure 8-15 Site framework and site areas

Note: For more information about the definitions and use of IBM Workplace Web Content Management site and site areas, refer to 5.3.8, “Site framework” on page 170 and “Site area” on page 172. Alternatively, refer to the IBM Workplace Web Content Management Information Center at:

http://pvcid.raleigh.ibm.com/ilwwcm/

As the pages are accessed, the dynamic cache continues to cache page fragments. The cached fragments can be viewed via the cache monitor. Use the following URL to access the cache monitor:

```
http://[portal_hostname]:[portal_port]/cachemonitor
```

For example:

```
http://wp51vm.ibm.com:9081/cachemonitor/
```

Archived

Figure 8-16 illustrates the cache monitor interface.

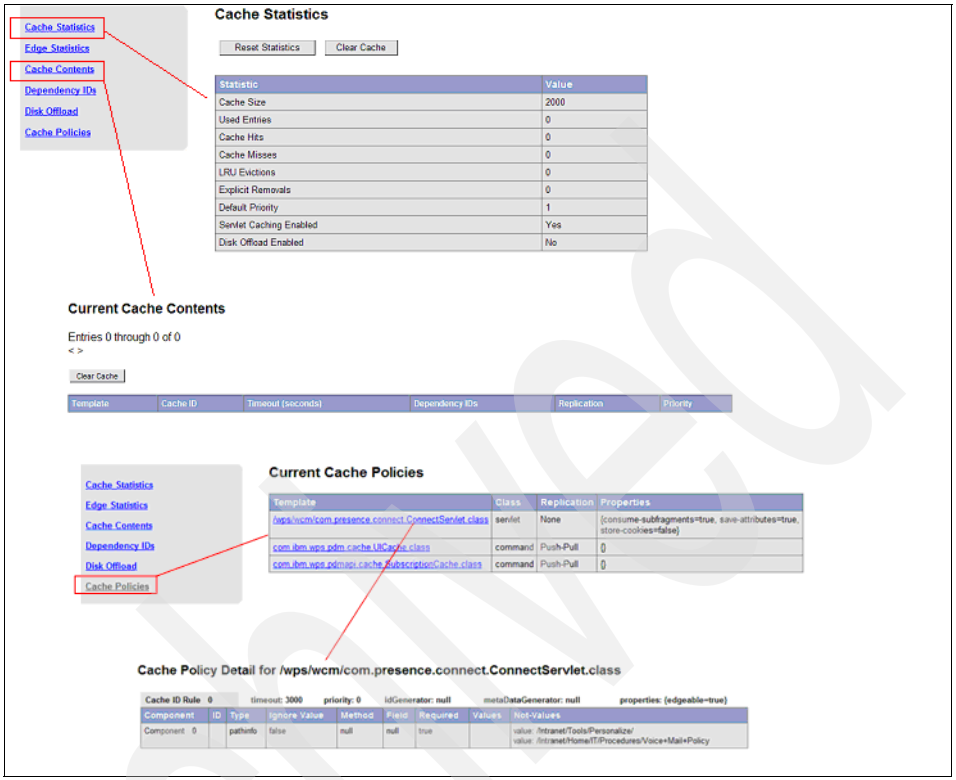


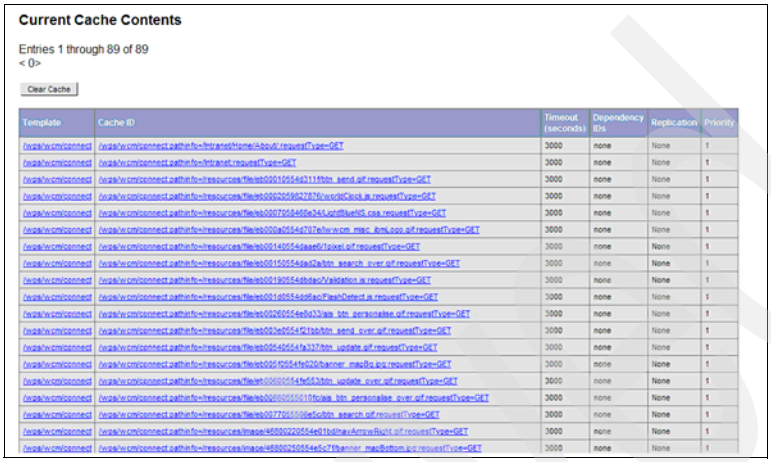
Figure 8-16 Cache monitor interface , including cache statistics and policies

Archived

For example, when a page is requested at the following address:

```
http://wp51vm.ibm.com:9081/wps/wcm/connect/Intranet/
```

Then, the cache contents look similar to that shown in Figure 8-17.



Current Cache Contents

Entries 1 through 89 of 89
< 0>

Clear Cache

Template	Cache ID	Timeout (seconds)	Dependency IDs	Replication	Priority
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/Intranet/Home/About/:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/Intranet:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00010554d3111f/btn_send.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00020598227f/wpdtClock.js:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00070568e8e34/LightBlueN5.css:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/000a0554d707e/w:wcm_mac_link.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00140554daae6/1pixel.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00150554da2a/btn_search_over.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00190554dbdac/Validation.js:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/001d0554dd6ac/FlashDetect.js:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00260554e8d33/aia_btn_personalise.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/002e0554f21bb/btn_send_over.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00540554fa337/btn_update.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00550554fe920/banner_macBg.jpg:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00560554fe553/btn_update_over.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00[illegible]10fc/aia_btn_personalise_over.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/file/eb/00770[illegible]8e5c/btn_search.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/image/46800220554e01bd/navArrowRight.gif:requestType=GET	3000	none	None	1
/wps/wcm/connect	/wps/wcm/connect/pathinfo=/resources/image/46800220554e5c7f/banner_macBottom.jpg:requestType=GET	3000	none	None	1

Figure 8-17 Current cache contents

Each fragment can be refreshed or invalidated individually and manually as is the entire cache.

Installing Edge Side Include

By enabling the Edge Component, contents such as images and JSPs can be cached by the Edge Component, which is installed typically on the front-line Web Servers. You must first install the DynaCacheEsi.ear file so that the Edge components and statistics can be managed via the Cache Monitor. Follow these steps:

1. Locate the DynaCacheEsi.ear file. It is located usually in the installedableApps directory of the [*WAS HOME*] directory (for example, C:\Program Files\WebSphere\AppServer\installableApps).
2. From the main WebSphere Application Server Administration Console, select **Applications** → **Install New Application**.
3. Choose **Local Path** and use the Browse button to locate the Cache Monitor install file.
4. Click **Next** to proceed.

Archived

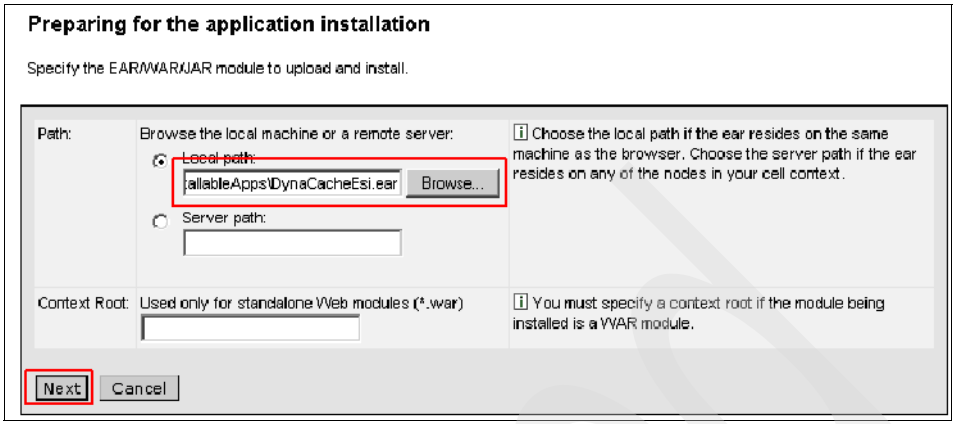


Figure 8-18 Installing the DynaCacheEsi.ear file

5. Accept the default settings as shown in Figure 8-19 on page 397.

Archived

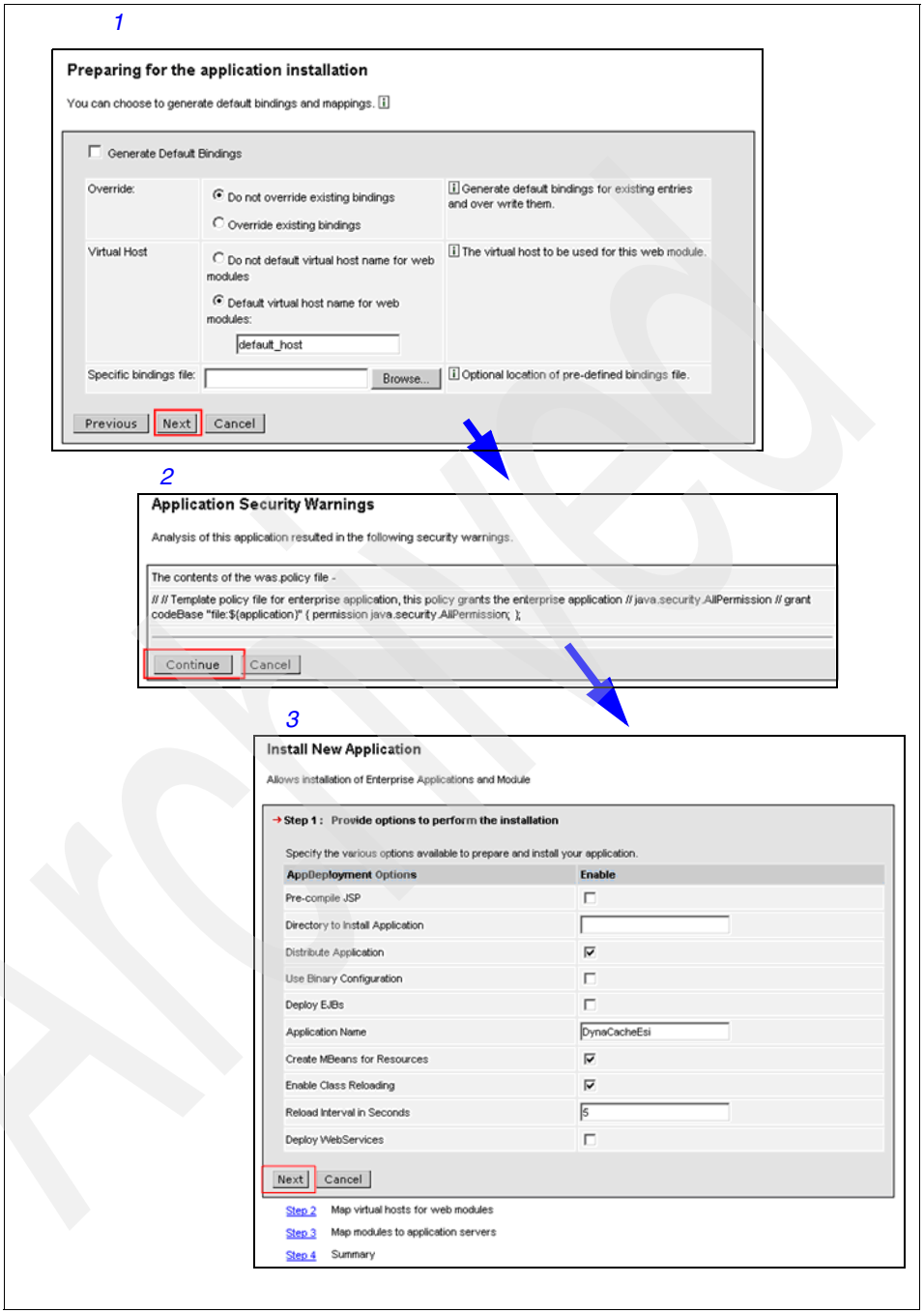


Figure 8-19 Installation default settings for DynaCacheEsi.ear

Archived

6. Map DynaCacheEsi.ear to a virtual host. In Figure 8-20, the `default_host` is used.

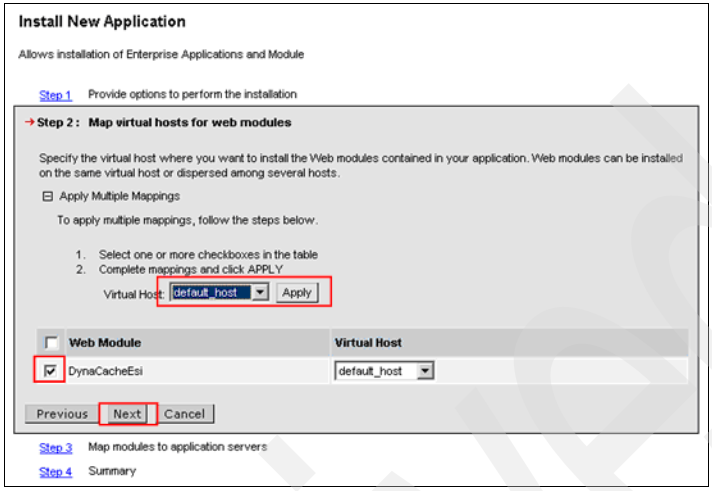


Figure 8-20 Mapping to a virtual host, selecting default_host

7. Map Web modules to the servers to be monitored (for example, WebSphere_Portal and server1), as shown in Figure 8-21.

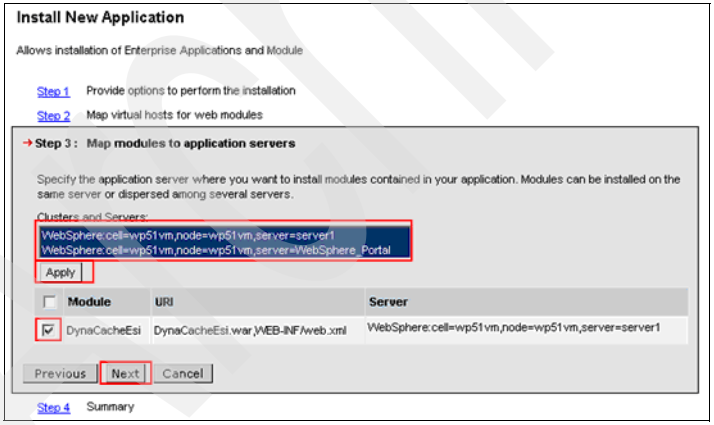


Figure 8-21 Mapping Web modules to servers

8. Click **Finish**, **Save to Master Configuration**, and **Save** as shown in Figure 8-22.

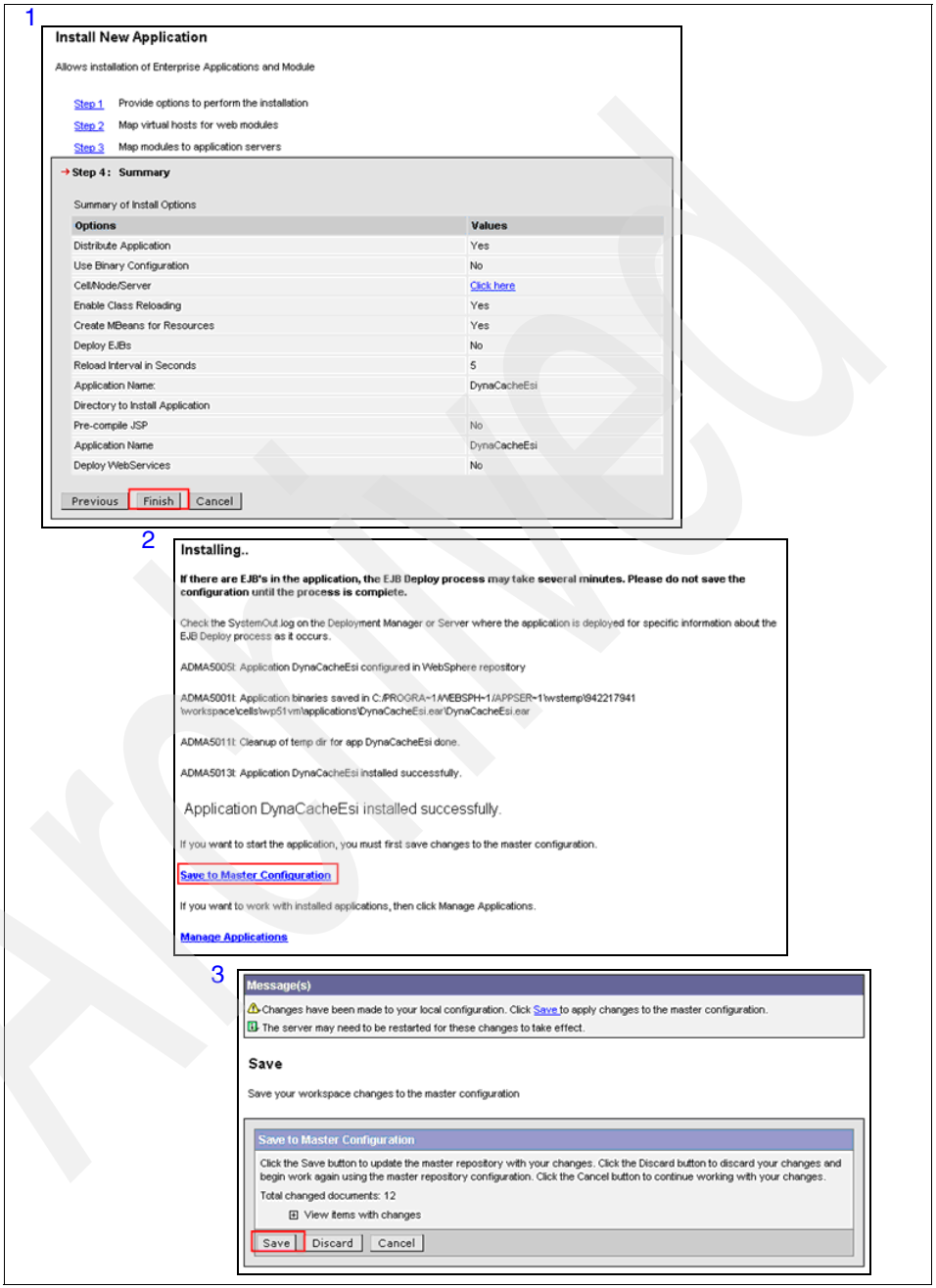


Figure 8-22 Finishing installation of DynaCacheESI.ear and saving

Archived

Enabling ESI components

You must enable the ESI components on the WebSphere Application Server plugin-cfg.xml. You must also amend the IBM Workplace Web Content Management cache policy to use the edge component.

This example caches fragments of the contents, such as images and JSPs by the Edge component (typically installed at the front line Web servers). To enable this component, the edge component is enabled on the plugin-cfg.xml, which is located in the [*WAS_HOME*]\config\cells directory, for example C:\Program Files\WebSphere\AppServer\config\cells.

The plugin-cfg.xml is generated by the server when the WebSphere Application Server plug-in is installed and the edge properties are there by default, as in the following:

```
<Property Name="EsiEnable" Value="true"/>
```

In addition, you should also set the following properties:

```
<Property Name="esiMaxCacheSize" Value="1024"/>
<Property Name="esiInvalidationMonitor" Value="true"/>
```

In this example:

- `esiMaxCacheSize` is the maximum size of the cache in 1 KB units. The default maximum size of the cache is 1 MB. If the cache is full, the first entry to be evicted from the cache is the entry that is closest to expiration.
- `esiInvalidationMonitor` specifies whether the ESI processor should receive invalidations from the application server. By default, it is set to `false`.

Example 8-9 shows the entries that plugin-cfg.xml should have.

Example 8-9 plugin-cfg.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Config ASDisableNagle="false" AcceptAllContent="false"
AppServerPortPreference="HostHeader" ChunkedResponse="false"
IISDisableNagle="false" IISPluginPriority="High" IgnoreDNSFailures="false"
RefreshInterval="60" ResponseChunkSize="64" VHostMatchingCompat="false">
   <Log LogLevel="Error"
Name="C:/PROGRA~1/WEBSPH~1/APPSER~1/logs/http_plugin.log"/>
   <Property Name="ESIEnable" Value="true"/>
   <Property Name="ESIMaxCacheSize" Value="1024"/>
   <Property Name="ESIInvalidationMonitor" Value="false"/>
   <VirtualHostGroup Name="default_host">
.
.
</Config>
```

Example 8-10 adds the `EdgeCacheable` property to the cachspec.xml to indicate that the fragment has to be cached on the edge.

Example 8-10 cachspec.xml

```
<?xml version="1.0" ?>
<!DOCTYPE cache SYSTEM "cachespec.dtd">

<cache>
   <cache-entry>
      <class>servlet</class>
      <name>com.presence.connect.ConnectServlet.class</name>
      <property name="store-cookies">false</property>
      <property name="save-attributes">true</property>
      <property name="consume-subfragments">true</property>
      <cache-id>
         <component id="" type="pathinfo">
         <required>true</required>
            <not-value>/Intranet/Search+Results</not-value>
         <not-value>/Intranet/Tools/Personalize/</not-value>
         <not-value>/Intranet/Home/IT/Procedures/Voice+Mail+Policy</not-value>
         </component>
         <timeout>3000</timeout>
         <property name="EdgeCacheable">true</property>
      </cache-id>
   </cache-entry>
</cache>
```

Archived

Archived

9

Advanced integration with WebSphere Portal

This chapter explains how to build more tightly integrated IBM Workplace Web Content Management and WebSphere Portal solutions. It includes the following sections:

- Overview of the WebSphere Portal integration
- Reasons for using Portal navigation
- Bridging the gap
- Beginning the implementation
- Implementing a custom display Portlet
- Managing links
- Using multilingual and personalized content display

Note: The integration methods that are described in this chapter are not a replacement for the publishing options that are outlined in previous chapters. The methods discussed here are an extension of the standard publishing. You can use them to implement more advanced functionalities.

© Copyright IBM Corp. 2006. All rights reserved.

9.1 Overview of the WebSphere Portal integration

The IBM Workplace Web Content Management implementation options that we have described so far come from a Web content management background. Thus far, we have used the WebSphere Portal framework mainly in the content creation process. We now discuss the options that you have to leverage the WebSphere Portal framework to its full extent.

From a developers point of view, WebSphere Portal is as a powerful framework on which to build Web applications. It provides 90% of the functionality that you need to build world-class Web applications. These features range from security, user-management, navigation, and integration APIs to Web content management. WebSphere Portal also removes from applications many of the design elements that are related to infrastructure and GUI support and allows the developer to focus on creating business functions in applications.

This chapter discusses how to build applications so that you mix and match content while also providing feature rich functionality. For example, you can have a product finder that interacts with your IBM Workplace Web Content Management content, or you can build Portlets that handle multi-lingual content display.

Before we jump into coding, however, we want to review the following important concepts:

- **Navigation**

  Chapter 5, “Key concepts and terminology” on page 141 describes the function of sites and site areas. Sites and site areas structure the content inside the IBM Workplace Web Content Management repository and build the site navigation.

  WebSphere Portal maintains a content structure as well. This structure is part of its *composition topology* and, therefore, is a a fundamental concept of WebSphere Portal. The Portal holds a tree-structure of its contents, which is called the *Content-Model*. Elements of the Content-Model represent pages with content, labels, or URLs. Each single page consists of layout information (containers) and content (Portlets). Portal pages are the containers for the Portals. One of the main purposes of this Content-Model is to build the Portal navigation. The composition topology is closely tied to the security model.

- **Review on Portlets**

  A *Portal* provides a consistent view of Portal applications (*Portlets*). The Portal allows the administrator to define specific sets of applications, which are presented to the user in a single page context. The Portlets themselves are more than simple views of existing Web content. A Portlet is a complete

application that has multiple states and view modes, plus event and messaging capabilities.

Portlets run inside the Portlet *container* of a Portal *server*, similar to a servlet running on an application server. The Portlet container provides a runtime environment in which portlets are instantiated, used, and finally destroyed. Portlets rely on the Portal infrastructure to access user profile information, participate in window and action events, communicate with other portlets, access remote content, look up credentials, and store persistent data.

The user interacts with Portlets through *Action-URIs*. By clicking a link inside the Portlet, the user can perform certain actions, for example, changing the display view. Internally, the Portlet stores persistent information about its current state. One Portal page aggregates many different Portlets at the same time, which is an important issue to consider when you design and work with Portals.

9.2 Reasons for using Portal navigation

The content-model is an important component in the WebSphere Portal architecture. To take full advantage of the Portal framework and its functionality, you should consider using the Portal Content-Model to build the site structure and the navigational schema. From an implementation point of view, using the Content-Model means the following:

1. You have to create one Portal page for every navigation entry in the site. Our sample site uses a head navigation and a left-hand navigation pane. (See Figure 9-1 on page 406.) Of course, you can build other navigation schemas as well. For pages that do not contain any content, you can use labels.

Archived

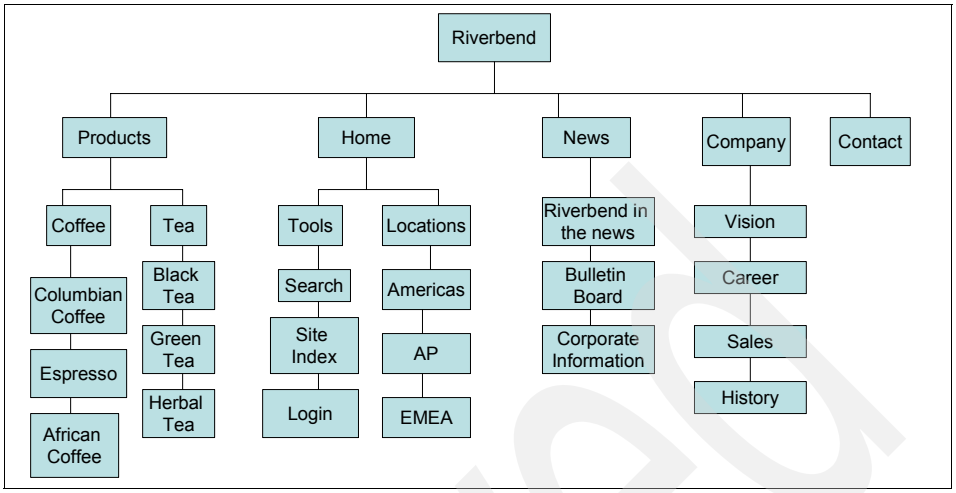


Figure 9-1 Each page and navigation element represents one Portal page

2. You need to create the Portal pages in a tree hierarchy that represents the hierarchy of the navigation, as shown in Figure 9-2.

Figure 9-2 Illustration of the Portal navigation hierarchy

3. You need to implement the navigation itself using the themes and skins of the Portal.

This implementation starts outside of the IBM Workplace Web Content Management environment. Thus, it is helpful for you to familiarize yourself with basic Portal administration and customization.

Archived

9.2.1 Understanding the layout of the Portal page

Before customizing your Portal site, it helps to understand the underlying structure of the Portal as it is determined by the Portal JSPs. The Portal page is composed of JSPs for screens, themes, and skins that are created typically by the Web designer of the Portal.

These JSPs reside in the corresponding /screens, /themes, /skins directories under the *was_profile_root*/installedApps/cellname/wps.ear/wps.war directory. Within this location, subdirectories for markup, locale, and client types are used to support Portal aggregation.

- Screens

  Screens represent the area of the Portal that typically displays portlets (that is, the Home screen) but can also display other content in its place (for example, a log in form or error message). Screens are selected from navigation icons in the theme.

- Themes

  Themes provide the navigation, appearance, and layout of the Portal, including colors, fonts, and images outside of the portlet content area (the Home screen).

- Skins

  Skins represent the border rendering around components, such as row containers, column containers, or portlets. Skins are installed independently from themes. However, the administrator can set a default skin for a theme.

The starting place for building the Portal page is Default.jsp in the /themes directory. For more information, see the Information Center at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/dgn_lyt.html

In the following sections, we assume that you have built the site, site areas, and input templates as described in Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253 and that you also have created some content. We now discuss how to integrate the IBM Workplace Web Content Management content with the WebSphere Portal Content-Model and that navigation that you have created.

Archived

9.3 Bridging the gap

How do the content documents know where they should get displayed inside of the Portal? Obviously, there is a gap between the WebSphere Portal Content-Model and the site and site areas that you have built. To bridge this gap, you can use *mapping schemas*. There are two types of mapping schemas: explicit schemas and implicit schemas.

9.3.1 Explicit mapping

Explicit mapping allow either the Portal administrator or the content authors to control where content is shown inside of the Portal. You have two options:

- You can have the administrator place the IBM Workplace Web Content Management local rendering Portlet on a Portal page. Then, you can configure which document from the IBM Workplace Web Content Management repository should be shown. Read 9.4.1, “Explicit mapping approach” on page 414 for more information about this option.
- You can allow the content author to assign a document to a Portal page. In this case, the content author selects a Portal page during the authoring process. The input template might offer the content author a list with all Portal pages from which to choose. Read 9.4.2, “Making the implementation more functional” on page 416 for more information about this option.

Figure 9-3 on page 409 illustrates the two different structures that you have to map together.

Archived

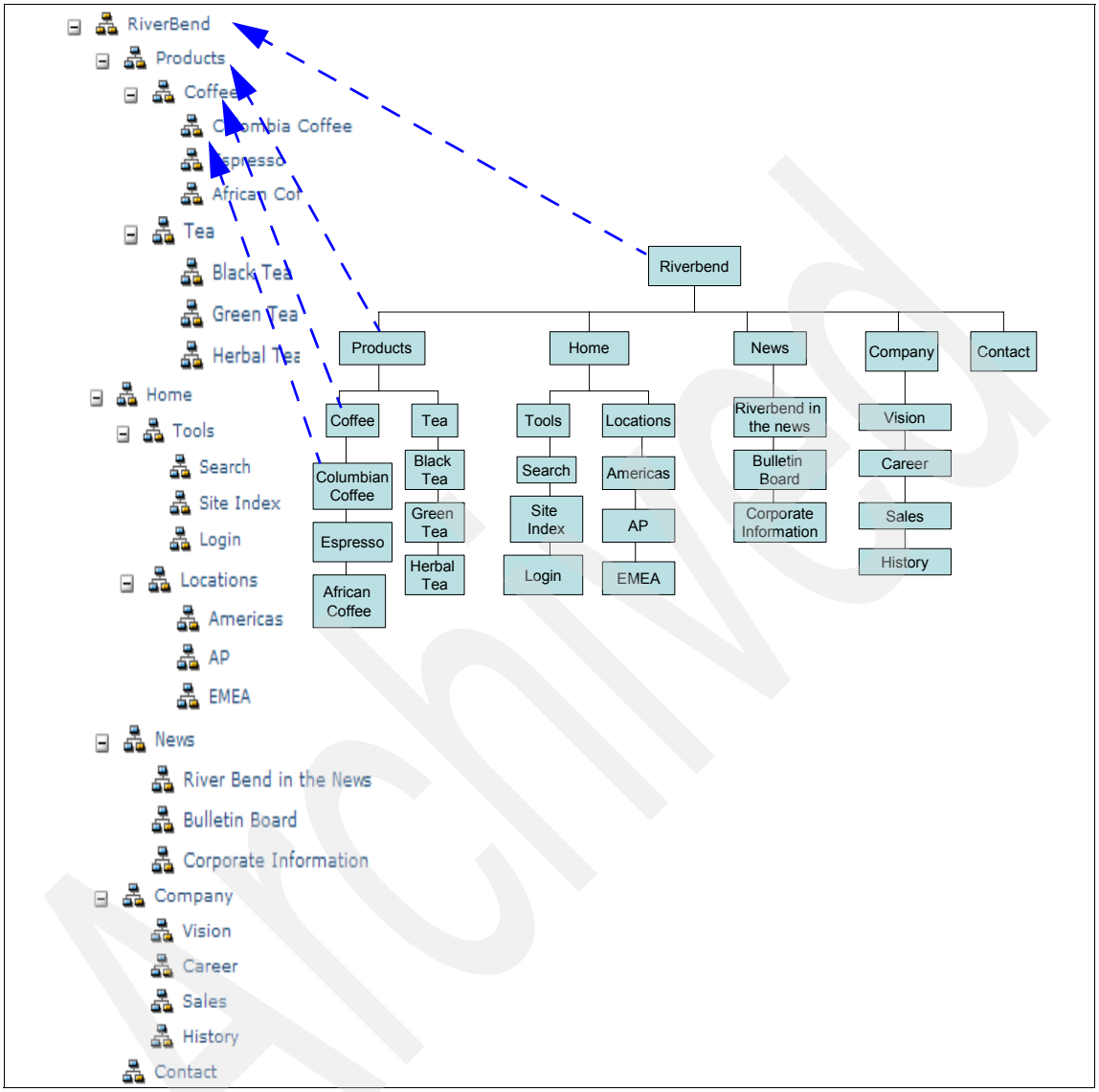


Figure 9-3 Site areas structure and Portal content model (page hierarchy)

On the left-hand side, the figure shows the site areas of IBM Workplace Web Content Management, and on the right-hand side, the figure shows the Portal page hierarchy.

Archived

The content author assigns a target Portal page explicitly for the content that is displayed. Figure 9-4 illustrates the resulting mapping schema.

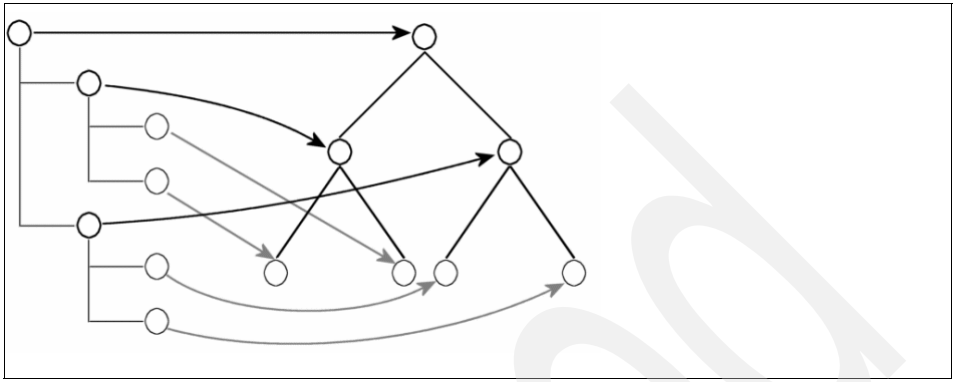


Figure 9-4 Explicit mapping between site areas and Portal page hierarchy

The mapping information is then stored inside of the content document.

9.3.2 Implicit mapping

With implicit mapping, you create rules that are based on a content taxonomy. For example, if `product=Coffee` then `Portal page= riverband.products.coffee`. The mapping rule is then evaluated by the Portlet code to decide what content document to display. A good way to work with implicit mappings is to use the rules engine of the Portals personalization engine. For more information, refer to the Information Center at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/pzn/pzn_demooverview.html

9.3.3 Choosing structures that make mapping easy

From our experience with customer engagements we have learned that we need to think about the IBM Workplace Web Content Management structure and Portal structure *before* beginning any implementation. To choose structures that make the mapping easy, you have the following options:

- **Use Portlet menus**

  Portlet menus allow a Portlet to add menu entries to the navigation tree of the Portal (Figure 9-5 on page 411). This functionality allows a custom IBM Workplace Web Content Management Portlet to extend the Portal navigation dynamically. This way, the IBM Workplace Web Content Management content can add its own sub navigation entries.

Figure 9-5 Example showing Portlet menu

The added Menu items are positioned automatically in the navigation tree after the page that contains the custom Portlet. The Portlet menu entries do *not* belong to the internal Portal content model.

Archived

Figure 9-6 illustrates Portlet menus that use the context of the River Bend site, as described in Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253.

Figure 9-6 Portlet menu example within the context of the River Bend site

The Portlet menu nodes contain a URL that either points to the Portlet itself or to an external page. When the menu nodes refer to the Portlet, the URL includes an action string on which our Portlet code can react (for example, to display a different IBM Workplace Web Content Management content). To add menu entries to the Portal navigation tree, the Portlet must implement the MenuProvider interface.

Note: The implementation of Portlet menus and dynamic menus is described very well in the WebSphere Portal Information Center at:

`http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/wpsbsmenu.html`

Archived

For more information and examples for using portlet menus, see the Javadoc in the com.ibm.wps.portlet.menu, com.ibm.wps.portlets.menu, and com.ibm.wps.portletservice.portletmenu packages. The Javadoc includes some very detailed examples of how to implement portlet menus.

- **Develop context navigation Portlets**

  Another way for the IBM Workplace Web Content Management Content to extend the navigation schema is via context navigation Portlets. In this case, the primary content can contain links with further reading or, based on a taxonomy, you query related content and display that inside of the context navigation Portlet, as shown in Figure 9-7.

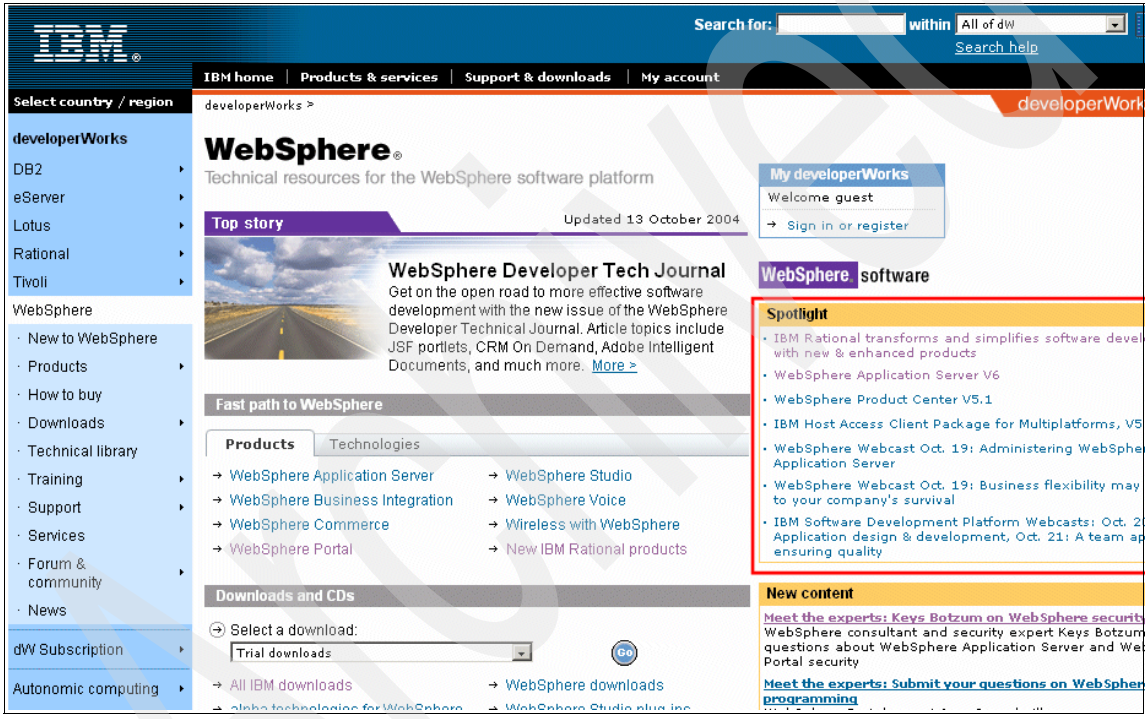


Figure 9-7 Example of a context navigation Portlet

Archived

Alternatively, Figure 9-8 illustrates a context navigator that uses the River Bend site, as described in Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253.

Figure 9-8 Contextual navigation within the context of the River Bend site

9.4 Beginning the implementation

Now that we have explained some of the concepts involved in the implementation, we discuss how to begin the implementation in this section.

9.4.1 Explicit mapping approach

With the explicit mapping approach, the administrator assigns one IBM Workplace Web Content Management content to every Portal page explicitly. Therefore, you need to place the IBM Workplace Web Content Management local rendering Portlet on a Portal page and configure it to display one content page. However, the local rendering Portlet does not support an edit mode. It only

has a configuration mode. Thus, you have to make copies of the Portlet so that it displays different content on different Portal pages.

Working in configuration mode

Figure 9-9 shows different variations of a Portlet as it is created, placed on a page, and accessed by users. Notice that the first two steps involve the use of persistent data.

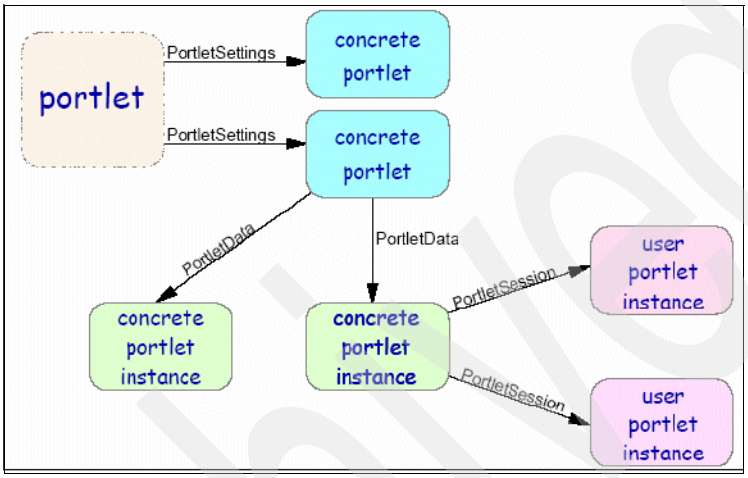


Figure 9-9 Variations of a portlet

- Config mode

  A concrete parameterization of a Portlet object is referred to as a *concrete* Portlet. The settings of a concrete Portlets can change at any time (for example, when administrators modify Portlet settings using the config mode of a Portlet). To have a Portlet with a different parameterization, you have to make a copy of the Portlet or implement an edit mode.

- Edit mode

  In edit mode, the Portlet is placed on a page by a user or an administrator. This action creates a concrete Portlet instance, which is a concrete Portlet parameterized by a single PortletData object. This configuration is done using the edit mode of the Portlet.

The scope of this configuration depends on the scope of the page on which the Portlet is placed:

- If an administrator puts a Portlet on a group page, then the edit configuration is shared for the group of users. This holds true for a group of users who have view access to the page. However, if users have edit access to the Portlet on a group page, then a new concrete Portlet instance is created for each user

Archived

that edits the Portlet. In this case, PortletData (Edit Mode) contains data for each user that edits the Portlet.

- If a concrete portlet is put on a user's page, the PortletData contains data for that user.

In our scenario, we assume that users have only view access to pages.

Conclusions for the explicit mapping approach

So, when considering the explicit mapping approach, do the following:

1. Make *n*-copies of the local rendering Portlet, where *n* is the number of Portal pages.
2. Implement a simple content display Portlet that has an edit mode and that stores the configuration information in the Portlet data.

For now, you can copy the local rendering Portlet, configure every instance, and then place them on the Portal pages. This approach might sound awkward. However, it allows you not just to use one local rendering Portlet per page but to have pages with different Portlets and page structure. So, you gain flexibility to leverage the Portal framework to build (for example, composite applications that go alongside with content from the IBM Workplace Web Content Management).

9.4.2 Making the implementation more functional

Now we discuss how to make the IBM Workplace Web Content Management and WebSphere Portal integration more functional. At the same time, we also want to reduce the administrative overhead that is involved with the use of the IBM Workplace Web Content Management local rendering Portlet. The solution is the implementation of a custom IBM Workplace Web Content Management display Portlet. (The actual implementation is discussed in detail in 9.5, “Implementing a custom display Portlet” on page 418.)

Before we talk about the implementation of the Portlet, let us first look at the authoring process. Having the administrator define what content is displayed on a Portal page might not be desirable. Usually, the content author is responsible for this action. To make this implementation work, we have to allow the content author to select the Portal page where content should be displayed. Thus, we need to extend the content creation template to allow the content author to choose from all available Portal pages. Internally, we store the unique-name of the page in case the page title changes.

This implementation can be done in several ways:

- You can maintain a list a all Portal pages manually, which implies an administrative overhead.
- You can use XMLAccess to query the Portal navigation. This query can be done periodically and might result in using a JavaScript library to populate the list.
- You can use the Model SPI (system programming interface) to get a list of all Portal pages.

In the following sections, we discuss more about the XMLAccess approach and the Model SPI approach.

Using the XML configuration interface (XMLAccess)

The XML configuration interface provides a batch processing interface for Portal configuration updates. It allows you to export or import an entire Portal configuration or parts of a configuration (for example, export specific pages to an XML file). You can then re-create the exported configuration from such a file on another Portal.

You access the XML configuration interface using a command line tool. This command line client is a small, separate program that connects to the server using an HTTP connection. It is, therefore, possible to configure the Portal remotely.

Tasks that you can perform with the XML configuration interface include:

- Exporting and importing the complete Portal configuration
- Exporting or creating sub-trees of the content-model
- Exporting or creating single page definitions
- Exporting or creating URL mappings

There are sample scripts available that ship with the Portal server in the \WebSphere\PortalServer\doc\xml-samples directory.

For example, ExportSubTree.xml allows you to export a sub-tree of the content hierarchy. The export result from this script is exactly what we would use to build the Portal page list.

Using the Model SPI

The Model SPI is an API that is used by Portlets and Portal applications to obtain read-only information about the Portal infrastructure. A number of models are available that can be queried, such as the Portal navigation model, content hierarchy, or the layout of a page.

Archived

Elements of the sub package com.ibm.portal.content define how content is represented in the Portal. The two main models of this package are the *ContentModel* and the *LayoutModel*. The ContentModel defines a tree structure for content elements (such as pages and labels). It is used to group these logically.

Note: You can find more information in the WebSphere Portal Information Center at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/dgn_modelovw.html

9.5 Implementing a custom display Portlet

Unfortunately, the local rendering Portlet cannot use the *content to Portal page* assignments. At this point, you have to implement a custom IBM Workplace Web Content Management content display Portlet. For information about how to implement a custom content display Portlet, see 12.5.1, “Developing a simple Portlet” on page 501.

The custom Portlet can now use the Portal page assignment that the content authors created. It just has to decide what content document it should display as its primary content. This can be decided in the following ways:

- The Portlet queries the IBM Workplace Web Content Management backend for a document that contains the same unique Portal page name as the Portal page on which the Portlet is currently placed. You can use the Model SPI to get the unique name of the current Portal page.
- What if it finds more that one content document that is assigned to the current Portal page? In this case, you have to implement a rule that selects the primary content to be displayed (for example, based on document fields or publishing dates).
- You could also query for the primary content based on a taxonomy.

9.6 Managing links

So far our solution depends on the Portal navigation for the user to navigate the site. We have not yet thought about links inside of our content or within our

presentation templates. First, there are different types of links that need to be supported:

- **Inside of the content documents**

  Typically, you want the content author to manage the links in the IBM Workplace Web Content Management. For example, while creating a new article, the author wants to link to a specific piece of content. When the user clicks the link, the content document is rendered on the same Portal page.

- **Across Portal pages**

  Sometimes the IBM Workplace Web Content Management document to which you have a link lives on a different Portal page. In that case, the link should take us to the correct Portal page and should show the content document to which we linked.

- **To Portal pages (pages with no Content Portlets)**

  In addition, we want to link to Portal pages that might have no content on them (for example, they host a feedback Portlet or other Portlet application).

Additionally, the user has a certain perspective and expectation of the functionality that links should support. For example:

- The user's navigational experience of a Web site — rather than an application — includes a *working back button.*
- Links should be able to be bookmarked, technically stable, and able to be sent in an e-mail.

Before we start to implement link management, let us first get a better understanding of how links work inside of a Portal framework.

9.6.1 Understanding links inside of the Portal framework

At first glance, the navigation inside of a Portal is comparable to the navigation of a simple Web page. This is true as long as the user navigates between Portal pages. However, the situation gets more complex as soon as the user interacts with a Portlet (for example, the local rendering Portlet). From the browser's point of view, interactions with a Portlet result in the loading of a new page. This observation is correct in that the URL changes. From a user's point of view, the user seems to be on the same page. Only the small area of the Portlet itself changes.

This usage pattern of the Portal becomes interesting as soon as the user presses, for example, the back button. Starting with WebSphere Portal Version 5.1, the Portal introduces a new concept called *navigational state handling*. This concept implies that users can now use the back button of the browser to navigate back through the recent history of the pages that they visited.

When a user navigates backwards using the back button, the Portal restores the views of pages that the user visited recently. This behavior affects the following aspects of the pages:

- The sequence by which the user navigated through the Portal and selected the pages.
- The expanded or collapsed state of the tree hierarchy that the navigation shows.
- The lateral scrolling position of the register tabs that users can use to switch between pages.
- The information of the user's default selection for a label. For example, this can be which page is selected for a label by default for that user. If the user selects a different page from that label, the Portal displays that page as the default page for that label from then on.
- The portlet window information that distinguishes between different instances of the same portlet on different pages.
- Modifications of the view or window state of a portlet, such as the minimized, maximized, normal, or default state. For example, if a user views a portlet in its default view state and then maximizes the portlet, clicking the back button returns the portlet to its previous state (that is, the portlet is displayed with its default size).
- The portlet modes, such as view, edit, or configure modes.
- The information as to whether a portlet is displayed in solo state, and which portlet is displayed in solo state.
- The information as to whether the Show tools icons are displayed, for example the wrench icon for configuring the portlet.
- If the portlet complies with the JSR 168 standard, the view state of the portlet is preserved as it is determined by the render parameters of the portlet.

  For details about state changes to JSR 168 compliant portlets, refer to *Using the Back button with JSR 168 portlets* in the WebSphere Portal Information Center at:

  http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/adbakbut.html#jsr168_portlets

  Thus, using the back button returns to a previously visited page, and then clicking a link on that page that has not been followed before correctly preserves any settings. This behavior is important to remember for the link management discussion later on.

Based on the requirements, our links need to be stateful (that is, they must contain all the information to recreate the currently displayed Portal page, which is what the user sees in his browser).

9.6.2 Implementing the links using 2-Phase links

To implement stateful links, we use the concept of 2-Phase links. 2-Phase links are links to a Portal page that contain the following information:

- In Phase 1, links change to a Portal page.
- In Phase 2, the custom IBM Workplace Web Content Management Portlet picks up information about the target document that it displays from an URL parameter.

To implement links, it might seem best to use user-friendly URLs. However, this approach might be not the best option.

User friendly URLs

URL mapping allows Portal administrators to create constant user-friendly URLs and to map them to Portal pages. As administrators create the URLs, they define human readable names for them. These names can be easily remembered and are, therefore, more user friendly. The self-defined URLs can be published externally and, thereby, can be made available to Portal users. For example, for the River Bend site, we could create a user-defined URL called *products/coffee* for the page on which we advertise our different coffee brands. This URL can then be appended to the Portal prefix of our Portal site (for example `http://www.riverband.com/wps/portal/products/coffee`).

Clicking such a mapped URL from outside of the Portal takes the user to the desired Portal page. Users can also combine several mapped contexts into the representation of a full valid URL, type that full mapped URL into the address field of the browser, and thereby get to the Portal page.

Although this linking sounds like the ideal concept to link across Portal pages, there is a drawback to using user-friendly URLs.

Portal URLs encode navigational state information about the Portal (for example, the user's currently selected page) and about the Portlets on a page (for example, the window state of the portlet) in a serialized form. Encoding navigational state information in the URL is used by the Portal server to support use of the browser's back button. Be aware that using user-friendly URLs interferes with the navigational state handling of the Portal and your Portlet applications — especially JSR168 Portlets that make use of rendering parameters. All previous navigational state information is lost if you use user-friendly URLs. Therefor, you should use user-friendly URLs *only* as a starting point into the Portal.

You can find more information about this topic in *Portal navigation and Back button behavior* in the WebSphere Portal Information Center at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html

This drawback is the reason why we implement link management differently. We use the Portal API to create working links properly. Technically, you can create links using the `CreateUrlCommand` class.

Note: The `CreateUrlCommand` class is the equivalent to the `URLGeneration` tag.

The `CreateUrlCommand` class allows you to create URLs to the following target resources:

- Pages or labels
- Portlets, with the option of adding render or action parameters (that's what we will need)
- Specific resources, such as an image file in the Portal
- Protected resources such that the URL is generated on the condition that the user has sufficient permissions

9.6.3 Implementing the link management

As a prerequisite for the link management, you also want to allow users to create links easily. Therefore, you need to implement a Link Helper Portlet. This Portlet allows the user to choose a content document from the IBM Workplace Web Content Management repository and then the Portlet creates the link. Here is how it works:

1. The Link Helper Portlet offers the user a list of all IBM Workplace Web Content Management content pages.
2. The IBM Workplace Web Content Management API queries for the linked to document and gets the unique-name of the Portal page to which the content is assigned.
3. The Link Helper Portlet shows the built link to the user, for example, `JavaScript:link2content([unique-name], [site are id]).`

The computed JavaScript links need to contain the information of the Portal page ([unique-name]) of the linked to document. Otherwise, you are not able to change the Portal page accordingly.

Important: Notice that we do not render the final links at this time. This rendering is done during the rendering phase of the custom IBM Workplace Web Content Management display Portlet. *This delay in rendering is necessary, because the Portal links contain the navigational state of the current user and are, therefore, user specific, which means that the links needs to be created on a per user basis.* This can only be done at run time.

Thus, we use a JavaScript function as a place holder to allow us to defer the link resolution to the rendering phase. By using JavaScript, we are also able to build the link list during the publishing process. This is very important, because building the link list involves content rendering and HTML parsing. Because these tasks are processing intensive tasks, we do not want to do them when the content is displayed.

9.6.4 Building the links

Initially, we have to identify what links are contained in our content documents. The issue is that these links can be embedded in Rich Text components or in the presentation template. To build a comprehensive list with all links, we have to render the content and then parse out all occurrences of the `link2content()` JavaScript function. With that information, we can then build a link list of all linked to documents and store this list in an IBM Workplace Web Content Management document field.

To implement this functionality, we have to extend the Customizable Template Portlet, as discussed in 13.2, “Using the Customizable Template Portlet” on page 535. After the users saves the content, we implement some code that renders the content, that then does the parsing, and finally that builds and saves the link list.

The link list is then used during the rendering phase of our custom IBM Workplace Web Content Management Portlet. There, we create the Portal links for the individual user using the `CreateUrlCommand` class.This class allows us to build links across Portal pages and to add URL parameters as well. The resulting links are stored in a string that represents a JavaScript array. Why a string? Well, we are still in the controller code of our Portlet at this time. So, we wrap the JavaScript array into a string to be used later on the JSP page that renders the content. The JavaScript array has two dimensions and can be used similar to a HashMap. As the key, we use the combination of unique page name plus document-ID. The value is the resulting link from the `CreateUrlCommand`. The string that contains the JavaScript array is then passed into the view.jsp and allows the `link2content` JavaScript function to resolve the links easily for us.

Important: The URL parameter is very important, because it allows us to implement phase 2 of our link management. We are going to add the document-id of the linked to document as an URL parameter.

Example 9-1 from the view.jsp shows how this implementation works.

Example 9-1 Sample from the view.jsp

```
<script language="JavaScript" type="text/javascript">
<!--
<%=portletRequest.getAttribute(WXMCustomPortlet.JS_LINK_ARRAY)%>
function link2content(unique_page_name, document_id)
{window.location.href=links[unique_page_name+ document_id];
}
//-->
</script>
```

Note: Notice that we have chosen this approach for performance reasons. Alternatively, you could render the content presentation at run time and replace the links inside of the HTML. This approach, however, is not good for the performance of the Portlet. Thus, we recommend that you not do it.

Important: It is best practice to do all the processing intense operations during the save and publishing cycle to ensure that you get optimal performance at run time.

9.6.5 Walking through of the linking process

This section contains a sample scenario that walks you through the link creation and link execution processes.

Scenario

We have two IBM Workplace Web Content Management documents. One is a news article, and the other is a product information (perhaps for a new coffee brand). The news article's primary Portal page is the news section, and the product information document belongs to the product section of the Portal.

The content author now wants to create a link to the product information from within the news article announcing the new product. Using the Link Helper Portlet, the content author chooses the product information as the destination document for the link. Internally, the Link Helper Portlet queries for the product information document and gets the unique-name of the assigned Portal page. It

then builds the JavaScript link. The computed URL might look similar to the following:

```
JavaScript: link2content([unique portal page name], [Workplace Web Content
Management document id])
```

When the content is saved, the unique Portal page name and IBM Workplace Web Content Management document ID are extracted and saved into a link list field. At run time, this link list is then used to create the user specific Portal links. If the user clicks the JavaScript link, phase one of the link management begins, and the Portal changes to the Portal page of the product area. Then, phase two is activated, and the custom IBM Workplace Web Content Management Portlet picks up the URL parameter and displays the content document to which we have linked.

9.7 Using multilingual and personalized content display

Earlier in this chapter, we discussed how a Portlet finds the primary content to display when a user navigates to a page and the Portlet has to decide what IBM Workplace Web Content Management content to display. We achieved this by querying the IBM Workplace Web Content Management backend for the unique-name of the Portal page. This concept worked well, but has potential for more.

Imagine that you have to write product documentation for a global corporation. You are responsible for the English versions of the document, but there are other colleagues who would do documentation in other languages. In addition, every product documentation uses different tailoring for sales, management, and technical audiences. How do you organize all these versions?

Well, the easiest way is to use folders in a shared file system. For each product, you create a folder to hold all the different versions that are available for that product. To keep the languages organized, you use descriptive extensions in the file names (for example _DE for German, _EN for English, and so forth).

So far, this approach sounds reasonably easy to manage. However, how do we organize the content in IBM Workplace Web Content Management. You can use site areas to logically organize language versions and personalized (tailored) versions in a simple way by using the site areas as a folder structure. Thus, every content page has its own site area and shares that site are with corresponding (language or personalized) content versions.

The custom Portlet still uses the unique-name to query the content. However, now this content might exist in alternative versions. From the content document, we can get the information regarding its corresponding site area. Then, from the

site area, we get access to all content documents (which are children of this site area). Finally, we can implement rules that decide what content should be displayed to the user, as follows:

- The user has defined German as his preferred language in the browser. The custom Portlet calls request.getLocale to get the user's language and uses this information to iterate through all the IBM Workplace Web Content Management content documents in the site area. The Portlet finds the site area that ends with the appropriate extension for the German language content. Alternatively, the Portlet can also get the user's language preference from the user's profile data or from `request.getHeader("Accept-Language" =` directly from the browser.

  What happens if the document is not available in the user's preferred language? In that case, the custom IBM Workplace Web Content Management Portlet displays the document in the site's default language.

  Alternatively, you could define that the Portlet always display content in the site's default language. If alternative languages are available, the Portal could offer a link to switch the language display. The information that an alternative language was requested then can be passed as a URL parameter to the Portlet.

- The user belongs to a the management user group. You might want to show different content on the Home page for members of the management team than for regular staff. In this case, you can take the same approach as you did for a user with a language preference other than the default language, except that you match the groups to which the user belongs with document attributes. Then, compare this information with an audience attribute inside of the IBM Workplace Web Content Management content document. You can use the PUMA SPI to access the profiles of a Portal user or group.

PUMA SPI

The PUMA SPI provides interfaces for accessing the profiles of a Portal user or group. It is used to find, create, modify, and delete users and groups. Also, profile information about the currently logged in user can be retrieved. You can find more information about this in *PUMA SPI overview* in the WebSphere Portal Information Center at:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/wpspuma.html

The PUMA SPI is a very simple but powerful concept that can be used for many different purposes.

> **Note:** Do *not* query the IBM Workplace Web Content Management backend every time a user makes a request. Doing so is usually not necessary and is not good for performance. Instead, consider a caching strategy, as described in Chapter 8, “Caching and pre-rendering considerations” on page 347.

If you are concerned about caching old values, implement a servlet that can delete entries from the cache. You can use the IBM Workplace Web Content Management document ID as the key. Then, in your input template, embed an image link to your servlet and pass it the document ID as an input parameter.

9.7.1 Ensuring link integrity

For any link management implementation, it is also important to think about link integrity. What happens when a linked document gets deleted or expires? Well, then the links in all documents that link to the deleted document become invalid. Therefore, it is necessary to check whether a document has references to it before deleting or expiring the document. This is a functionality that is usually implemented in the IBM Workplace Web Content Management engine.

Because we have implemented the link management ourselves, we need to implement some link checking mechanism on our own also. Remember that during the save operation of an IBM Workplace Web Content Management document, we build a link list with references to all documents to which the content links. We can now use this link list to build a link checking process. We can then run this link checking on a periodic schedule to check for broken links. The result might be a list of documents that contain broken links that are displayed in an administrative Portlet. (A detailed description about how to implement link checking is outside of the scope of this book.)

9.7.2 Other linking functionalities

Other functionalities that might be useful for a custom IBM Workplace Web Content Management Portlet include:

- Linking to a non-content page

  Sometimes, you might want to link to a Portal page that hosts just a Portlet or Portal application but that contains no content. The easiest way for content authors to link to a non-content page is to do the following:

  a. Create an empty document in IBM Workplace Web Content Management.
  b. Assign the empty document to a Portal page (Portal unique name).
  c. Link to the document.

> **Note:** Even if there is no Content Portlet on the page, the `URLGeneration` class changes the Portal page for us. We do not recommend that you use a user-friendly URL. Refer to “User friendly URLs” on page 421 for more information.

- Making the Content Portlet hide if no URL parameter or Portlet Data default is present

  This option is useful if you want another Portlet to control the behavior of the custom IBM Workplace Web Content Management Portlet (for example, a product finder).

  Go to `http://www.aofoundation.org` and click **Innovations**. You see a skeleton that functions as a content navigator. If you click a bone segment, it displays product thumbnails from the IBM Workplace Web Content Management repository. If you then click a product, the hidden content Portlet becomes visible and the skeleton hides instead.

  You can implement this behavior by configuring the custom IBM Workplace Web Content Management Portlet to display no default content. Instead, it looks for the URL `document-id` parameter. When the user clicks a product thumbnail, the skeleton Portlet places the document ID into the URL. The custom IBM Workplace Web Content Management Portlet then picks up this information and displays the desired product document. The hiding of the skeleton Portlet is implemented in the same way — just the other way around. The `skeletonHide=true` parameter is present in the URL.

  You cannot use Portlet messaging to implement this behavior because you cannot reproduce the same result from a bookmark or link in an e-mail.

Archived

10

Search functionality for IBM Workplace Web Content Management

Managing your Web content in a structured environment such as IBM Workplace Web Content Management is a good step towards proper management of the content. However, the content is not very helpful if visitors cannot locate the information that they need. This chapter discusses the different ways that you can implement the search functionality so that visitors can find content easily.

This chapter includes the following sections:

- Providing multiple techniques to search content
- Using the IBM Workplace Web Content Management search module
- Using the IBM Workplace Web Content Management API
- Using the WebSphere Portal search engine
- Using external and third-party search products

© Copyright IBM Corp. 2006. All rights reserved.

10.1 Providing multiple techniques to search content

If your site is large or if it contains a wide variety of content, visitors expect the site to provide one or more techniques for searching the content. IBM Workplace Web Content Management 5.1 provides multiple techniques for searching its content repository, which include the following:

- IBM Workplace Web Content Management integrated Search Module
- IBM Workplace Web Content Management API
- WebSphere Portal Search
- Third-party search products

Each search option has its advantages and disadvantages. In many situations you need a combination of these capabilities to meet your specific requirements.

> **Tip:** As you read through this chapter, do not think of each option as exclusive of the others. Look at ways that each is suited to solve a specific problem.

10.2 Using the IBM Workplace Web Content Management search module

You can use the IBM Workplace Web Content Management search module to create a basic search function on content objects, keywords, and categories that are stored in an IBM Workplace Web Content Management content repository. In addition, you can combine queries with `and/or` operators to create a more specific query.

The search module's functionality is independent from the data repository type that you use to store your IBM Workplace Web Content Management objects. Thus, no matter which content repository you choose (Cloudscape, DB2, or DB2 Content Manager), the search results are the same.

At a high level, here is how the IBM Workplace Web Content Management search module works:

1. The site developer creates a search form and formats how the search results appear using HTML. The developer saves the HTML code in a combination of IBM Workplace Web Content Management components, such as presentation templates, HTML components, and content items.
2. A site visitor performs a search by submitting one or more words in the search form. The search module compares the words against the search index and lists content pages that match the search words as search results.

10.2.1 Security and the search module

The search module takes into account the security settings on content during a search. A search returns only the content that the requestor is allowed to see. Searching external content (for example, content in DB2 Content Manager) is not supported unless you include the external content in your Web page using the Federated Content Component.

10.2.2 The search module index

When you enable the search module, the search module creates a search index. This index is refreshed at regular intervals based on the settings in the <*WP_Root*>/wcm/config/aptrixsearch.properties configuration file. The configuration file contains a variety of settings, including four that directly effect the search index, as shown bold in Example 10-1.

Example 10-1 Search Index related settings from aprtixsearch.properties

```
# the directory where IBM Workplace Web Content Management search
# should store its hit indexes
search.directory=D:/WEBSPH~1/PORTAL~1/wcm/ilwwcm/search

# the hit index refresh interval in minutes, optional
refresh.interval=120

# the hit index refresh times, optional
# note that these times must match the format of your locale.
#refresh.time=18:00:00;04:00:00

# the username to use when indexing ilwwcm (optional)
# choices are Anonymous, Administrator or a specific user or group common name.
ajpepageserver.user.name=wpsadmin
```

The `refresh.interval` and `refresh.time` parameters control how often the index is updated. Content does not appear in a search result until it is added to the index. However, the indexing process can effect the performance of your IBM Workplace Web Content Management server. The more often your content changes, the more frequently you should index.

The IBM Workplace Web Content Management search engine index is stored on the file system of the IBM Workplace Web Content Management server at <*WP_Root*>/wcm/ilwwcm/search/hitIndex.dat.

Note: The search application returns `no-results` until it creates the index.

Archived

If necessary, you can limit what text is indexed by creating a search filter using a presentation template. For example, you can exclude text from navigators and menus.

For complete details about configuring IBM Workplace Web Content Management and building the necessary assets to use the IBM Workplace Web Content Management search module see Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253, and in particular, 7.24.3, “Modifying a presentation template for the Search page” on page 329.

10.3 Using the IBM Workplace Web Content Management API

The IBM Workplace Web Content Management API provides access to the content within the content repository. From a technical perspective, it is feasible to use this API to build a search engine for IBM Workplace Web Content Management content. However, one should not embark on a process simply because it is possible. There are issues that you need to address if you build your search engine using this method, including the following:

- Consider the caching method that you use and the performance of the site. A large site requires a very strong caching algorithm to give the search engine a reasonable performance.
- Consider the security that the site uses. You need to ensure that content is visible only to the appropriate people.

If the standard IBM Workplace Web Content Management search module does not meet your needs, you can build your own search engine. However, due to the level of effort that is required to write a good search engine, we do not recommend this approach. Instead, we recommend that you look at existing search products that can search your IBM Workplace Web Content Management content repository.

Archived

10.4 Using the WebSphere Portal search engine

The WebSphere Portal search engine is made up of several components:

- Administrative Portlets that allow you to define search parameters, for example, the search scope.
- A crawler component that works similar to an Internet spider and that indexes the site.
- Search collections that represent index information and allow searching capabilities.
- Portlets that allow a user to search inside of a search collection.

In configuring the WebSphere Portal search engine, you can specify a starting point for the crawler. The crawler then follows links from one page to the next, so that it can gather information and learn about a Web site infinitely. The WebSphere Portal search engine crawler then stores the gathered information inside a collection for searching. This crawl can happen to two types of content sources:

- WebSphere Portal sites
- Regular Web sites

The difference between these two types of content is the way that the crawler follows links inside of the sites.

10.4.1 Crawling a WebSphere Portal site

When crawling a WebSphere Portal site, the crawler fetches and indexes all pages with Portlets to which it has access rights. This way, you enable visitors to search those pages. You can define which Portlets on which pages are searchable by granting the required access permissions to the crawler user.

To help the crawler index a WebSphere Portal site in an efficient way, the Portal provides the crawler with a *seed-list* page. This page contains information about all Portal pages and Portlets that the crawler should index.

Technical background to the seed-list indexing method

If you browse the content model of your Portal, you find a Portal page named wp.search.seedlist. This page hosts a Portlet named Manage Seed List. The seed-list Portlet builds an XML structure that contains references to all Portal pages and Portlets. To build the XML structure, the seed-list Portlet traverses the content model for the Portal. The WebSphere Portal search engine crawler then uses this XML data to index the Portal site.

The XML structure looks similar to that shown in Figure 10-1.

```
<PRE>
   <portal>
      <page page_id="6_0_IP">
         <title>d3BzYWRtaW4vSGVsbG9Qb3J0bGV0&#10;</title>

         <description></description>

         <url>L3dwcy9teXBvcnRhbC8hdXQvcC9rY3htbC8wNF9Taj1TUH1rc3N5MHhQTE1uTXowdkOwWV9RanpL&#10;TE4OZzNjUWJKZOZpTy1wR1FBVzk5WDRfODNGVD1BUDJD

         <portlet portlet_def_id="3_0_IP" portlet_id="7_0_LU">
            <title>SGVsbG9Qb3J0bGV0IHBvcnRsZXQ=&#10;</title>

            <description></description>

            <url>L3dwcy9teXBvcnRhbC8hdXQvcC9rY3htbC8wNF9Taj1TUH1rc3N5MHhQTE1uTXowdkOwWV9RanpL&#10;TE4OZzNjUWJKZOZpTy1wR1FBVzk5WDRfODNGVD1BUI
         </portlet>
      </page>

      <page page_id="6_0_CI">
         <title>d3BzYWRtaW4vVGVzdFBvcnRsZXQ=&#10;</title>

         <description></description>

         <url>L3dwcy9teXBvcnRhbC8hdXQvcC9rY3htbC8wNF9Taj1TUH1rc3N5MHhQTE1uTXowdkOwWV9RanpL&#10;TE4OZzNjUWJKZOZpTy1wR1FBVzk5WDRfODNGVD1BUDJD

         <portlet portlet_def_id="3_0_FL" portlet_id="7_0_FM">
            <title>VGVzdFBvcnRsZXQgcG9ydGxldA==&#10;</title>

            <description></description>

            <url>L3dwcy9teXBvcnRhbC8hdXQvcC9rY3htbC8wNF9Taj1TUH1rc3N5MHhQTE1uTXowdkOwWV9RanpL&#10;TE4OZzNjUWJKZOZpTy1wR1FBVzk5WDRfODNGVD1BUI
         </portlet>
      </page>
```

Portal page unique name

Page title base64 encoded

ID of first Portlet on the page

URL to open the Portlet in SOLO mode

Figure 10-1 XML structure returned by the seed-list Portlet

All the information that is rendered by the seed-list Portlet is base64 encoded. The XML structure contains URLs to all Portal pages as well as URLs to display every single Portlet in *SOLO* mode, meaning without the surrounding navigation. The WebSphere Portal search engine uses the title and description information as metadata in the indexing process. The WebSphere Portal search engine crawler now traverses the XML tree, calls all the links, and indexes the resulting pages.

A drawback to the seed-list indexing method is that the crawler does not follow any links inside of a Portlet because the links could be actions potentially. By following the links, the crawler would execute these actions, which could result in unexpected content creation or other unwanted data processing. In WebSphere Portal 5.1, the crawler cannot distinguish between navigational and action links. Therefore, it does not follow links inside of a Portlet at all. You should keep this in mind for your IBM Workplace Web Content Management content indexing.

In the case where you want to index a regular Web site, you can provide the crawler with a starting URL and with information about how deep you want to index the site. The depth is defined by the `Levels of links to follow` parameter. This parameter tells the crawler at what level to stop indexing. If you index regular Web sites, you should also consider including a filter that tells the

crawler *not* to follow links that are outside of the current domain. Otherwise, it might find a link to an external site and might start to index that site.

10.4.2 Considerations for using the WebSphere Portal search engine

Before using the WebSphere Portal search engine to search IBM Workplace Web Content Management content inside of our WebSphere Portal site, you should consider the implications of using this method.

For example, imagine that you have created a Portal page that holds an IBM Workplace Web Content Management navigation portlet and a display Portlet, similar to that shown in Figure 10-2.

Figure 10-2 Portal page with an IBM Workplace Web Content Management navigation and display Portlet

In this example, the WebSphere Portal search engine uses the information from the seed-list Portlet to index this Portal page. The seed-list XML structure contains two references for this portal page — one for each Portlet on the page.

Each entry allows the crawler to open each Portlet in SOLO mode and to index the resulting HTML. The result is only two entries in the search collection. As title,

it uses the title of the Portlet, not the IBM Workplace Web Content Management document title. The crawler also does not follow links inside of the Portlets. So, there is no way for the crawler to index all the IBM Workplace Web Content Management content documents that are linked in the IBM Workplace Web Content Management navigator.

10.4.3 Crawling a regular Web site

The next logical option to index the IBM Workplace Web Content Management content is to have the crawler index the Portal as a regular Web site. In this mode, the crawler actually indexes all of the IBM Workplace Web Content Management documents, with the following limitations:

- All IBM Workplace Web Content Management documents that are displayed on the same Portal page have the same metadata. Thus, all documents appear under the name of the Portal page inside the index. Why? The WebSphere Portal theme takes the title of the Portal page and renders it as a `<title>` tag in the head section of the HTML. This tag is what the crawler picks up and uses as the page title for the index. Because all of the content documents are shown on the same page, *they all appear under the same name*, presenting a significant limitation of this search method.
- The crawler follows all links. However, it cannot distinguish between navigational links that display additional content and Portlet actions. This behavior can result in unexpected actions inside Portlets. You need to be very careful if you choose this indexing method for Portal sites.

10.4.4 Crawling using a custom method

This section discusses ways to overcome the limitations of these search methods. The idea is to build a custom seed-list Portlet for IBM Workplace Web Content Management content. In this section, we call this Portlet the *Site-Summary Portlet*. The Site-Summary Portlet uses the IBM Workplace Web Content Management API to build a list of all IBM Workplace Web Content Management content documents of the site. In addition, it manipulates the metadata that is picked up by the WebSphere Portal search engine crawler. This method allows us to pass the IBM Workplace Web Content Management document title and other metadata to the crawler.

Building the Portlet

To creating the custom crawling approach within the portlet, follow these steps:

1. Use the IBM Workplace Web Content Management API to build a list of all IBM Workplace Web Content Management documents, as described in Chapter 12, “Working with the application programming interface” on page 469.
2. Create a link-list to all the IBM Workplace Web Content Management content documents by leveraging the 2-Phase linking concept, as described in 9.6.2, “Implementing the links using 2-Phase links” on page 421. This method is based on the Portal's `URLGeneration` tag. It allows us to create links to a Portal page and to add additional information to the URL.

   We use the 2-Phase linking as follows:

   a. The `URLGeneration` tag creates a link to the portal page that hosts the IBM Workplace Web Content Management local rendering Portlet.

   b. We add the information about what IBM Workplace Web Content Management document that we want to display to the URL, as shown in Example 10-2.

Example 10-2 Code that shows the URL generation

```
<wps:urlGeneration contentNode=”[PORTAL_UNIQUE_PAGE_NAME]”>
   <wps:urlParam name=WCM_GLOBAL_CONTEXT”
value=”[WCM_CONTEXT_ROOT_PATH]+[WCM_SERVLET_PATH]+[CONTENT_PATH]”/>
   <wps:urlParam name=”TITLE” value=”[IWWCM_DOCUMENT_TITLE”/>
   <% genURL = wpsURL; %>
   <a href = “<% wpsURL.write(out); %>”>[LINK_LABEL]</a>
</wps:urlGeneration>
```

   c. This information is passed in the `WCM_GLOBAL_CONTEXT URL` parameter that tells the local rendering Portlet what IBM Workplace Web Content Management document to display.

> **Note:** Make sure that your local rendering Portlet is configured to listen to the passed URL parameters. In the config-mode specify the following:
>
> `Receive Links From: 'Other portlets and this portlet'`

   The resulting links look similar to this:

```
…?WCM_GLOBAL_CONTEXT=/wps/wcm/connect/Products/Coffee/Columbia Coffee
```

   d. We add a `TITLE` parameter that passes the IBM Workplace Web Content Management document title to the portal theme.

3. To make use of the `TITLE` parameter in the URL, change the head.jsp of the Portal and pick up the `TITLE` parameter to render it to the HTML head as the `<title>` tag, as shown in Example 10-3.

Example 10-3 Page title contained in the document title

```
<head>
   <title><%request.getParameter["TITLE"]%></title>
   ...
</head>
```

Then, if the WebSphere Portal search engine crawler follows the link, the page title is the IBM Workplace Web Content Management document title.

If the WebSphere Portal search engine crawler follows the links in the link-list, the crawler goes to the Portal page with the IBM Workplace Web Content Management rendering Portlet. In addition, the URL contains the ID parameter that tells the IBM Workplace Web Content Management rendering Portlet what document it should display.

4. Create a Portal page and place the Site-Summary Portlet on it. To allow the WebSphere Portal search engine crawler to access the page easily, create a user-friendly URL for the page.
5. Add a content source to the search collection that uses the following parameters:
   a. Crawl source type: `Web site`
   b. Collect documents linked from this URL: `[your user-friendly URL]`
   c. Levels of links to follow: `1 level`
6. Run the WebSphere Portal search engine crawler to build the index.

   If you now use the Search Center or Search & Browse Portlet to search the IBM Workplace Web Content Management search collection, you find that the search result shows the proper IBM Workplace Web Content Management document titles. If you click a link, the link takes you to the correct Portal page and also opens the IBM Workplace Web Content Management document that you requested.

Archived

10.4.5 Using multiple Portal pages to display content

If you use multiple Portal pages to display IBM Workplace Web Content Management content, you have to implement some additional logic inside the Site-Summary Portlet. In this case, the following scenarios are possible:

- You have multiple Portal pages that host the IBM Workplace Web Content Management navigation and local rendering Portlet.

  In this case, you need to build a map that tells the Site-Summary Portlet which Portal pages displays which IBM Workplace Web Content Management site area. Table 10-1 shows an example.

Table 10-1 Example map for the Site-Summary Portlet

This site area	Maps to this Portal page
Riverbend/Products	Products
Riverbend/Tea	Tea

  The map information is then used together with the `URLGeneration` tag to create links to the corresponding Portal page.

- You have used the explicit or implicit mapping schema approach from 9.3, “Bridging the gap” on page 408.

  In this case, you just re-use the mapping code from the custom IBM Workplace Web Content Management display Portlet to calculate the corresponding Portal page for your IBM Workplace Web Content Management content.

10.4.6 Considering outside search engines

You should also consider how external search engines, such as Goggle, locate your IBM Workplace Web Content Management content. To make it easy for external search engines to index your site as well, you can present the Site-Summary Portlet to them for indexing too. The technique to achieve this is called *cloaking*. Cloaking is the process of delivering one version of a page to one user and a different version of a page to another user, such as a search engine. There are three ways of cloaking:

- IP delivery, where the IP addresses of spiders are recognized at the server and handled accordingly.
- User-Agent delivery, where the spiders' User-Agents are recognized
- A combination of IP delivery and User-Agent delivery.

Archived

A simple way to implement cloaking is to use the mod_rewrite module on the IBM HTTP server. The re-write module can look for the User-Agent and can re-direct Google to the Site-Summary Portal page.

10.5 Using external and third-party search products

This section highlights other tools and technologies for searching IBM Workplace Web Content Management content beyond the built-in IBM Workplace Web Content Management Search feature and the WebSphere Portal search engine. While IBM Workplace Web Content Management and WebSphere Portal provide search capabilities within their standard features, IBM and other vendors offer search specific products with more comprehensive features and functions.

Third-party search products include:

- IBM WebSphere Information Integrator OmniFind™ Edition

  http://www-306.ibm.com/software/data/integration/db2ii/editions_womnifind.html

- Apache Lucene

  http://lucene.apache.org/

- Verity Ultraseek

  http://www.verity.com/products/ultraseek/

Each search product has different features, functionality, support, IBM Workplace Web Content Management integration, and cost. Furthermore, every IBM Workplace Web Content Management project has a different set of goals and requirements. Which third-party product you choose is based upon the specific requirements for your project and organization.

Archived

10.5.1 Key considerations when select a third-party search product

While there are numerous factors that can enter into a decision on which product best suits your needs, most Web content project have a common set of core considerations, as follows:

- How content is delivered

  One of the most important issues that you need to analyze is where you will deliver your content:

  - Stand-alone Web site (via an IBM Workplace Web Content Management rendering server)
  - WebSphere Portal (via rendering portlets or custom portlets)
  - Stand-alone Web site **and** WebSphere Portal

  If you are delivering content in a single environment, the goal would be to find a search engine that performs best for that scenario. If you want to deliver content both stand-alone and within WebSphere Portal, then you have a more complicated search requirement. Some search products work well in all scenarios, while others work well in less complicated scenarios.

- Return on investment (ROI)

  At some level, all business decisions have to look at cost. There is a cost to implementing a search engine, and there is a cost associated with people not finding the content for which they are looking on your site. When considering which search engine you need for your environment, consider the following:

  - Cost

    The purchase price of the software can be significant. Most products require some form of maintenance for ongoing support and upgrades. Make sure that you understand all of the costs that are associated with the initial purchase and the ongoing maintenance of a given solution.

  - Time to implement

    When you purchase a search product, you need to install, configure, and implement the solution to work in your environment. This time cost is often difficult to define, but underestimating this effort can lead to flawed ROI numbers.

When you take these (and other) factors in total, it is possible that the ROI is not high enough to invest in a third-party search product.

Archived

Note: For further details on IBM WebSphere Information Integrator OmniFind Edition, see *WebSphere Information Integrator OmniFind Edition: Fast Track Implementation*, SG24-6697, which is available at:

http://publib-b.boulder.ibm.com/redpieces/abstracts/sg246697.html?Open

Archived

11

Personalization rules for IBM Workplace Web Content Management

In WebSphere Portal implements rule-based content delivery through a feature known as *Personalization*. Personalization lets Web sites customize content automatically for each user. Personalization can recognize a specific user based on a profile. It can also determine characteristics of a user based on previous purchases, products, or pages viewed and then select content that is appropriate for that profile.

This chapter focuses on the new personalization integration with IBM Workplace Web Content Management that is available in WebSphere Portal 5.1.01. This new feature allows Personalization rules to be created based on the basic attributes of IBM Workplace Web Content Management content. It includes the following sections:

- Overview of the personalization feature
- Installing and configuring WebSphere Portal 5.1.01
- Enabling WebSphere Portal personalization

© Copyright IBM Corp. 2006. All rights reserved.

Important: This functionality requires WebSphere Portal 5.1.0.1. Although this chapter describes how to do update to this version, you might also want to refer to the following:

http://www-12.lotus.com/ldd/doc/uafiles.nsf/docs/wcm5101/$File/wcm-5-1-0-1-releasenotes.htm

11.1 Overview of the personalization feature

This section summarizes some basic terminology and concepts that are needed to implement the basic scenario in this chapter. You can find complete information about the rich set of capabilities that the new personalization feature provide in the WebSphere Portal Information Center, which is available at:

http://www-306.ibm.com/software/genservers/portal/library/index.html

Figure 11-1 illustrates the components that we use in our scenario.

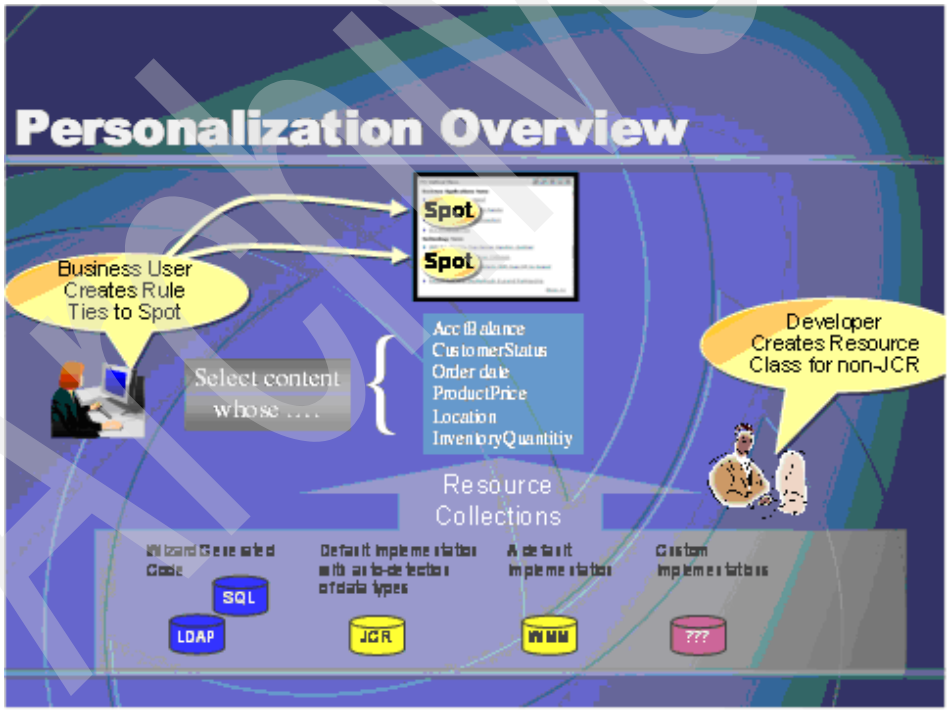


Figure 11-1 Overview of the personalization features

A Portlet can have a designated *spot* for rule-based content. For each spot, a rule is selected. All rules are created based on the resource attribute definition. These attributes are defined in a *resource collection*. The resource collection is a set of Java classes that retrieve content that is requested by the rule from the defined repository source. This backend source might be LDAP or SQL databases, the Portal Java Content Repository (JCR), WebSphere Member Manager, or any other custom collection that is coded for the personalization API. The JCR resource collection is provided with WebSphere Portal as well as WebSphere Member Manager. There are wizards available in WebSphere Application Developer and RAD6 that guide developers in creating classes for other LDAP and relational database resources. A guide of how to create these resources is described fully in the WebSphere Portal Information Center.

For our scenario, we use the personalization sample resources and the JCR. We do not create any classes using the wizard.

Three types of rules are used to target select content to a certain group of users:

- The *profile* rule can be made on any user resource. A user's profile is best described as a named group that is based on user characteristics. For example, a user is profiled as a *premium* user if the amount of the user's mortgage is over a certain amount. Another example could be that if a user's order amount is over a certain amount, then that use is profiled as a *gold* user. In addition, an employee could be profiled as non-management if the employee's `isManager` attribute is set to `false`.
- A *select* rule chooses content based on the attributes of the content. For example, the GoldOffers rule selects products whose retail price is greater than a certain amount. With IBM Workplace Web Content Management content, you can also select content with a specific category. For example, a NonManager rule gets content that is created with the HRNews template that has a category of non-manager.
- A *binding* rule ties the profile and select rules together. So, premium users see PremiumOffers, gold customers see GoldOffers, and non-manager employees see NonManagement news. You do not have to coordinate the naming of the profile with the select rule name. However, this convention helps when creating the binding rule.

Any of these rules can be tied to a content spot. To enable your environment to construct these rules, you need to install and configure personalization and IBM Workplace Web Content Management.

11.2 Installing and configuring WebSphere Portal 5.1.01

To upgrade an existing WebSphere Portal 5.1 system, you use the Portal Update Installer and the fix pack. Download the WebSphere Portal Version 5.1.0.1 cumulative fix and the WebSphere Portal update installer from the following Web site:

http://www.ibm.com/software/genservers/portal/support/

The following files are required:

- Cumulative fix file: WP_PTF_5101.jar
- WebSphere Portal update installer: PortalUpdateInstaller.zip

Use the instructions located in the WebSphere Portal Information Center to install this fix:

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html

Search on `5.1.0.1` to find the information that you need.

11.3 Enabling WebSphere Portal personalization

If you want to use personalization rules on IBM Workplace Web Content Management content, then you must configure IBM Workplace Web Content Management to *shadow* some of its content in the Java content repository. This repository is based on the JSR 170 standard and is the base repository direction for IBM Enterprise Content Management.

11.3.1 Changing the connect.cfg file

To write rules for IBM Workplace Web Content Management content with WebSphere Portal personalization, you must enable a JCR shadowing within Web Content Management. To do this, follow these steps:

1. Stop the IBM Workplace Web Content Management Application.
2. Edit the connect.cfg file that is located in the /PortalServer/wcm/config directory.
3. Find the following line (located before the `<LogManager>` section), and change the value to `true`.

   ```
   JCRRepository value=true />
   ```

4. Restart IBM Workplace Web Content Management.

Note: If LDAP has been enabled, you need to edit the following values within the `<JCRConnector>` element in the connect.cfg file to include the following values:

```
<DefaultUsername value="uid=wpsadmin,o=default organization"/>
<DefaultPassword value="wpsadmin"/>
```

You can find further information about LDAP settings in the LDAP user registry topic in the WebSphere Portal Information Center.

Note: In i5/OS® installations, IBM Workplace Web Content Management files are located in the WebSphere Portal user root directory, not in the /PortalServer/wcm/ directory. For example, for WebSphere Application Server 5.1.1.3, you can find the files in this directory:

/QIBM/UserData/WEBAS51/Base/instance name/PortalServer51

For WebSphere Application Server 6, you can find the files in this directory:

/QIBM/UserData/WebSphere/AppServer/V6/Base/profiles/instance name/PortalServer51

11.3.2 Using personalization rules on IBM Workplace Web Content Management Content

To write rules you need to define a resource collection. To create a resource collection, do the following:

1. Log in to WebSphere Portal.
2. Navigate to the Personalization Workspace.
3. Click **New** → **Resource Collection**.
4. Enter a name for the collection (for example, *IBM Workplace Web Content Management Resource Collection*).
5. Specify the following settings under Standard Resource Collection, as shown in Figure 11-2 on page 448:
   - Resource Class: `com.aptrix.pluto.pzn.resources.WCMResource`
   - Manager Class: `com.aptrix.pluto.pzn.resources.WCMResourceManager`
   - Domain Class: `com.aptrix.pluto.pzn.resources.WCMResourceManager`

   Be sure that there are no trailing spaces in these fields.

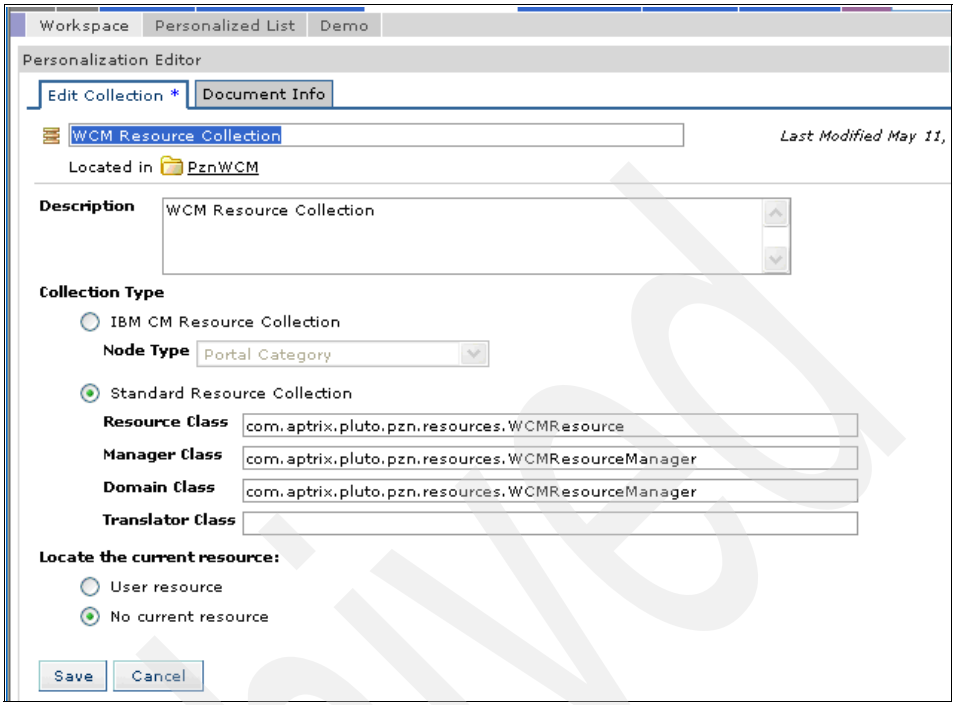


Figure 11-2 Specify settings for the Standard Resource Collection

6. Select **Save**.

11.3.3 Populating the JCR with existing content information

Since the integration of IBM Workplace Web Content Management, content for rules relies on the IBM Workplace Web Content Management rule attributes to be stored in the JCR. Because this feature is optional, you need to populate the JCR with your IBM Workplace Web Content Management content attributes initially. After you populate the JCR and configure IBM Workplace Web Content Management to shadow content in the JCR, any new content is shadowed automatically.

To execute this one-time step run the following:

```
http://yourserver/wps/wcm/connect?MOD=JCRModule&action=populate
```

This module created in this command is a requirement for the resource collections definition that is described in the previous section, because the classes are used to access the JCR.

Archived

Your IBM Workplace Web Content Management content is now shadowed in the JCR, and you are ready to write some rules.

11.3.4 Creating a personalization rule

To create a personalization rule, write a select rule that retrieves all the content that is created using the DeptNews authoring template. You can find a guide to creating this content at:

http://www.ibm.com/developerworks/websphere/library/techarticles/0503_smit/0503_smit.html

1. Log in to WebSphere Portal.
2. Navigate to the Personalization Workspace.
3. Select **New** → **Rule**.
4. Enter a name for the new rule (for example, `Get WCM Content`).
5. Select **Resource.Attribute**.
6. Select the Resource Collection that you created earlier.
7. Select the Web Content Management metadata that you would like personalization to use as the search criteria (for example, Template), as shown in Figure 11-3.

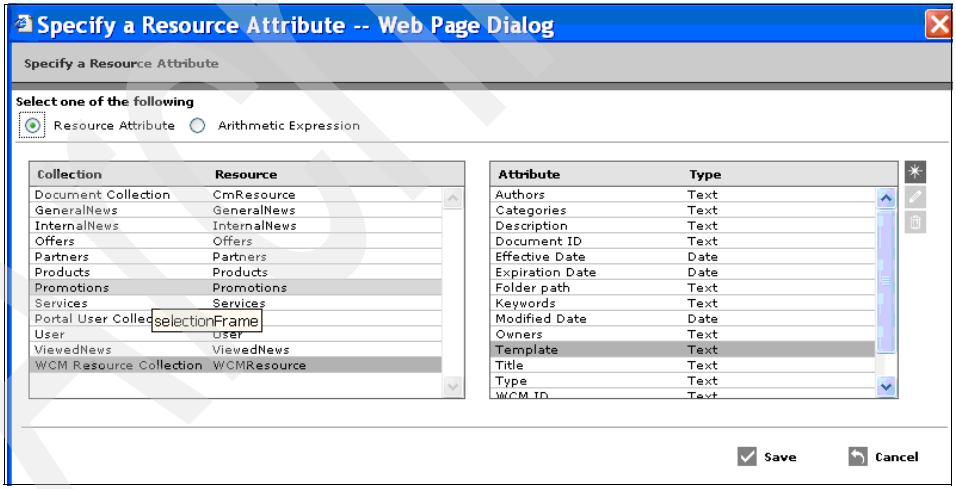


Figure 11-3 Specifying a resource attribute

Archived

8. Select **Value** and enter in a value for which you want to search (for example, your authoring template name), as shown in Figure 11-4.

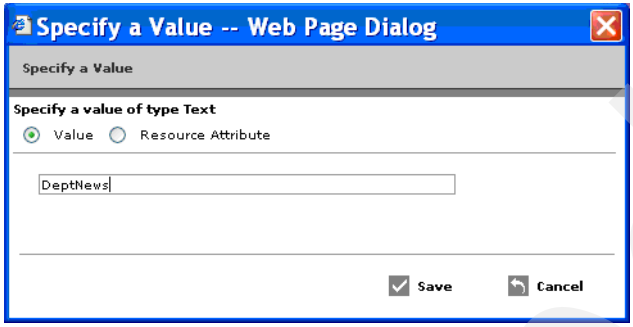


Figure 11-4 Entering a value to search for

9. Select **Save**.

10. To see that content is returned from this rule, select the Preview tab, as shown in Figure 11-5. The preview tab displays the IBM Workplace Web Content Management internal ID of the content and shows you that you do have content returned from your rule.

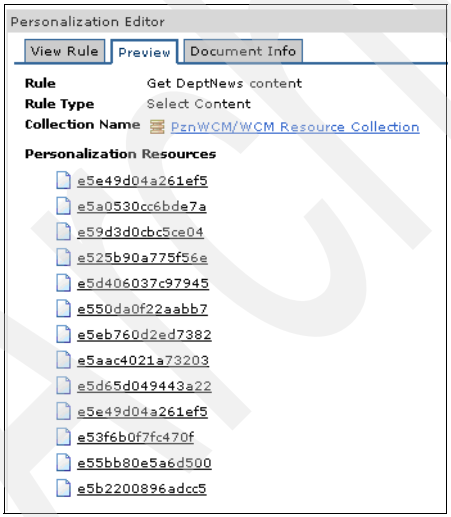


Figure 11-5 The Preview tab in the Personalization Editor

This Personalization Rule can now be displayed in a Personalization List Portlet, referenced in an IBM Workplace Web Content Management component, or included within a Content Spot on a Portal page.

Archived

11.3.5 Configuring the Personalized List portal

The Personalized List Portlet is installed with WebSphere Portal and can be added to any Portal page. A portal page is configured within the Personalization page with this list Portlet already on it when WebSphere Portal is installed. It has many configuration options that enable you to use a rule to show content quickly. There are two configuration modes:

- One that configures the Spot information
- One that configures the Display Options

Initially the Configure Spot is displayed, as shown in Figure 11-6.

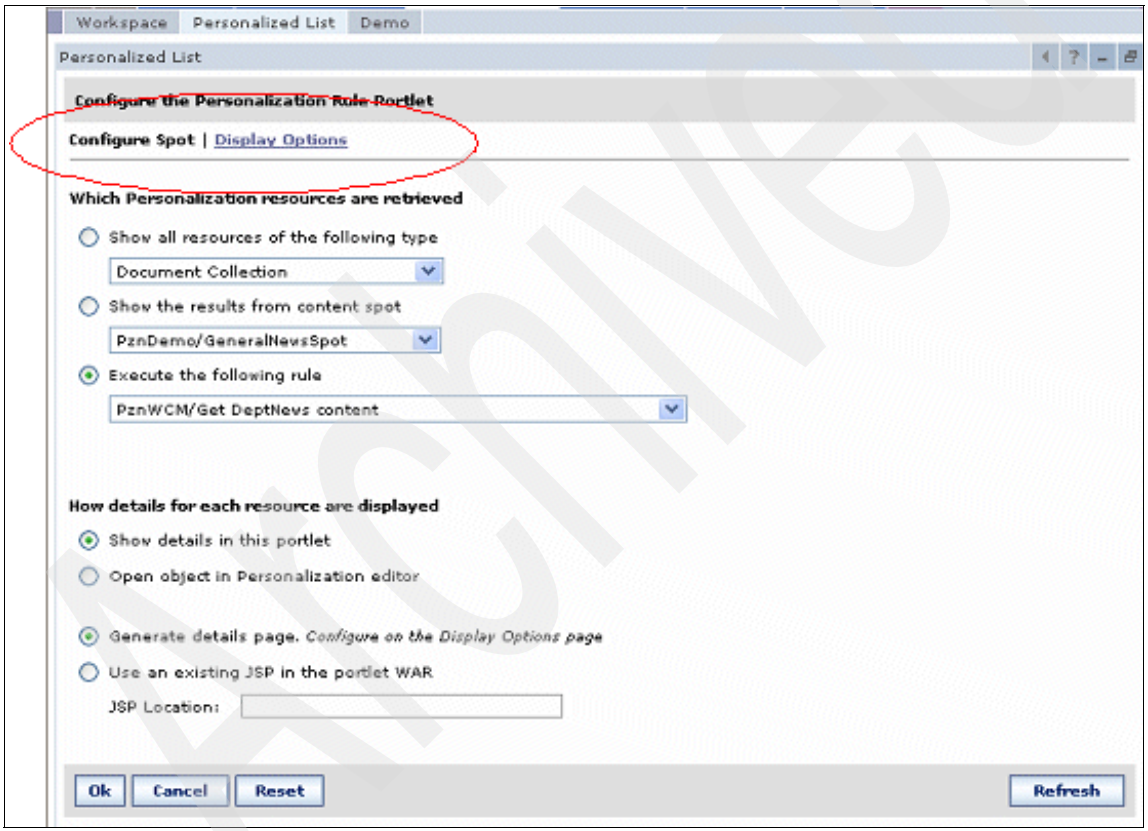


Figure 11-6 The Configure Spot configuration mode

Archived

To configure the Personalization List Portlet, do the following:

1. Select the rule that you just created as the rule to execute (see Figure 11-7).
2. Click **Display Options**.
3. Choose the Fixed Title attribute for the Title Attribute and choose the Fixed Description attribute for the Detail Attribute.
4. Select the Fixed Folder Path for the Categorization Attribute. This groups the content under the site area structure.
5. Click **OK**.

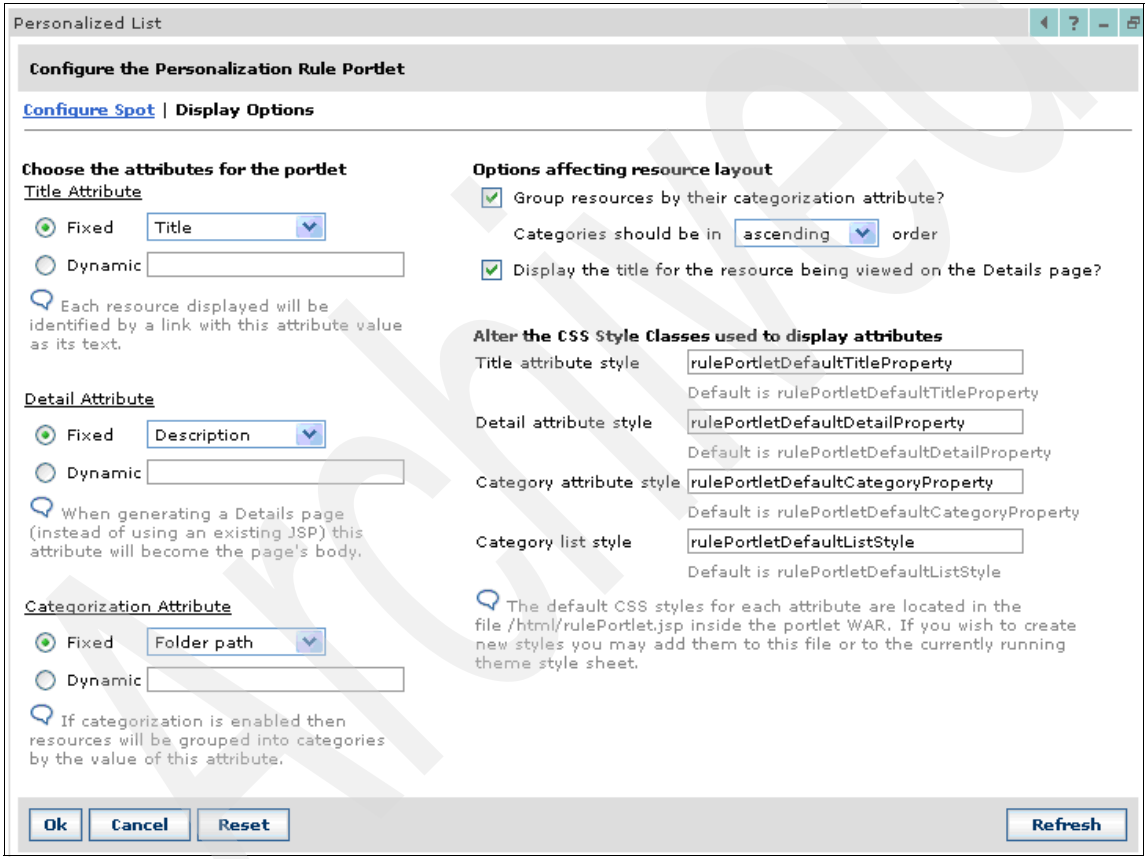


Figure 11-7 Display options for the Personalized List Portlet

Archived

Your rule content is returned (as shown in Figure 11-8).

Figure 11-8 Personalized List content

When you select a title, you get the detail attribute displayed for that title. Note that this list does not use the site area and presentation template to render the content. Instead, it uses just the attribute that you defined in the Portlet configuration.

11.3.6 Including rules on presentation template

A IBM Workplace Web Content Management presentation template lets you format the content as you need it. The new personalization component has been added to the list of IBM Workplace Web Content Management library components. You can select this component when configuring the IBM Workplace Web Content Management local rendering Portlet, or you can reference it in a presentation template.

Archived

To reference the personalization component in a presentation template, do the following:

1. In the IBM Workplace Web Content Management author user interface Portlet, select **New** → **Personalization Component** and click **OK**.
2. In the Identification dialog box, enter a name and a description for the personalization component (Figure 11-9).

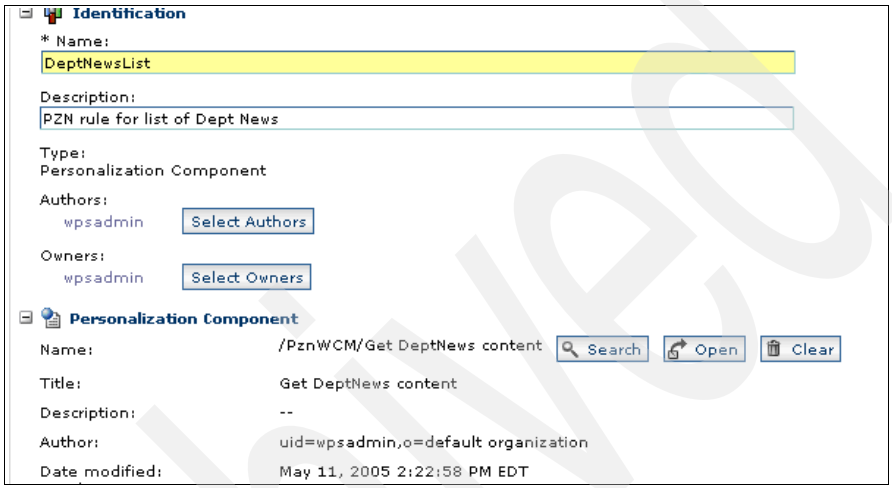


Figure 11-9 New personalization component

3. Select search and find the new personalization component that you created, as shown in Figure 11-10.

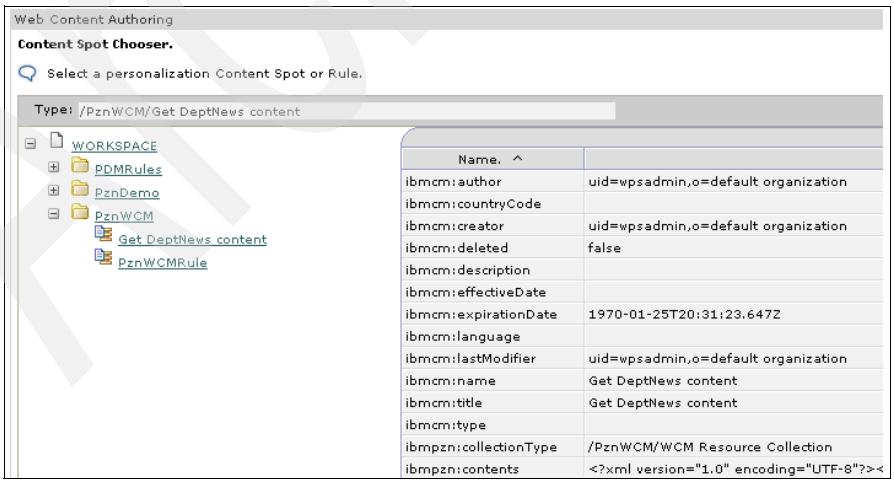


Figure 11-10 Listing of Personalization Components

4. Key in the heading and detail HTML tags as shown in Figure 11-11.

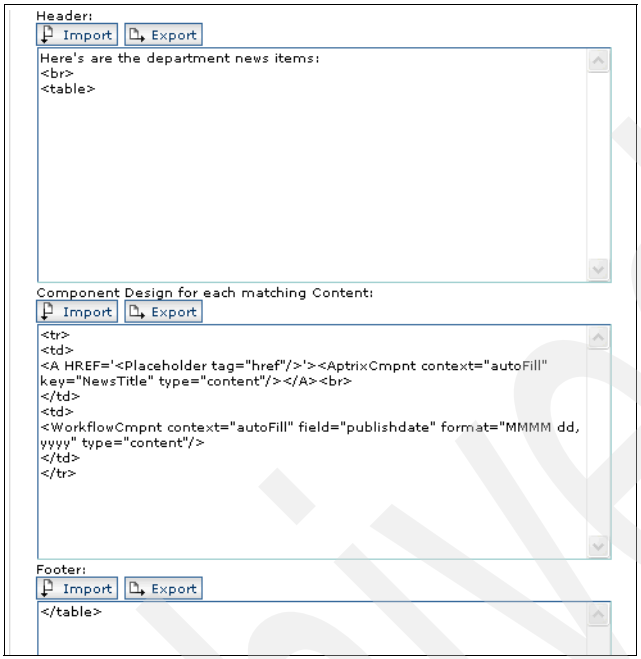


Figure 11-11 Heading and detail HTML tags

5. Save and close.

Next, you imbed this component in a presentation template as follows:

1. Copy an exiting detail presentation template (such as NewsDetail). Save this presentation template as *NewsDetailPzn* (Figure 11-12 on page 456) after including the following tag:

```
<AptrixLibCmpnt name="DeptNewsList"/>
```

Archived

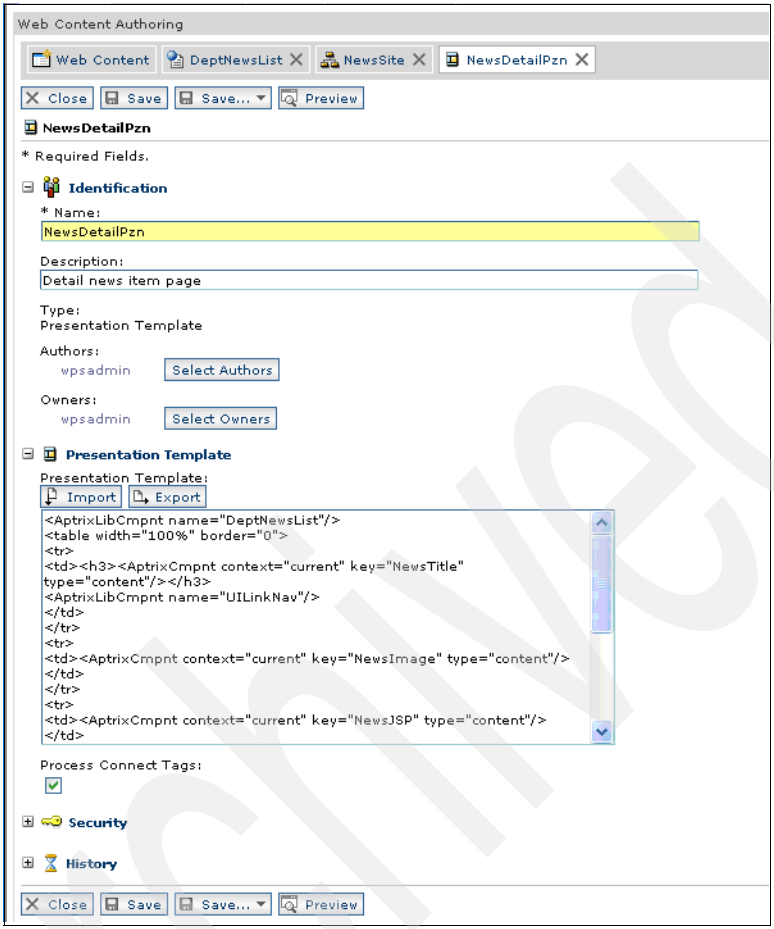


Figure 11-12 Saving the NewsDetailPzn presentation template

Archived

2. Click **Preview** to see the presentation template with the site area that matches your content (as shown in Figure 11-13).

Figure 11-13 Previewing the new presentation template

You have now used a rule to render IBM Workplace Web Content Management content. Next, you add a user profile and binding rules to direct specific content to certain users based on the attributes of the user.

11.3.7 Creating rules

A profile rule needs a user resource. You can use the portal resource, but it is more common to use some sort of database repository that contains employee or customer information. The personalization demo provides a user resource and populates a database with user attributes. The portal user is connected with these attributes via a translator class.

Archived

To create rules:

1. Create a *profile* rule named *UserType*, as shown in Figure 11-14. This rule classifies the user as a manager if the role attribute is set to manager.

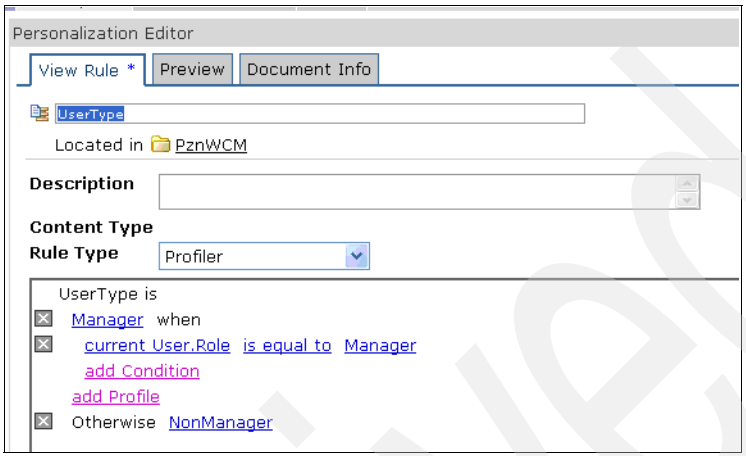


Figure 11-14 Creating a profile rule

2. Create a *select* rule that selects new content for managers called *Manager News*, as shown in Figure 11-15.

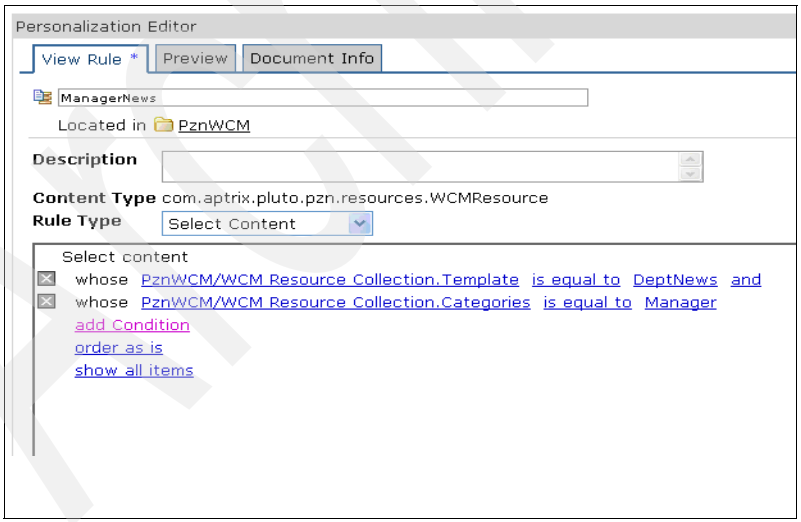


Figure 11-15 Creating a select rule

Archived

3. Exclude the manager items from the Dept News for non-managers by changing the rule that to match the rule shown in Figure 11-16.

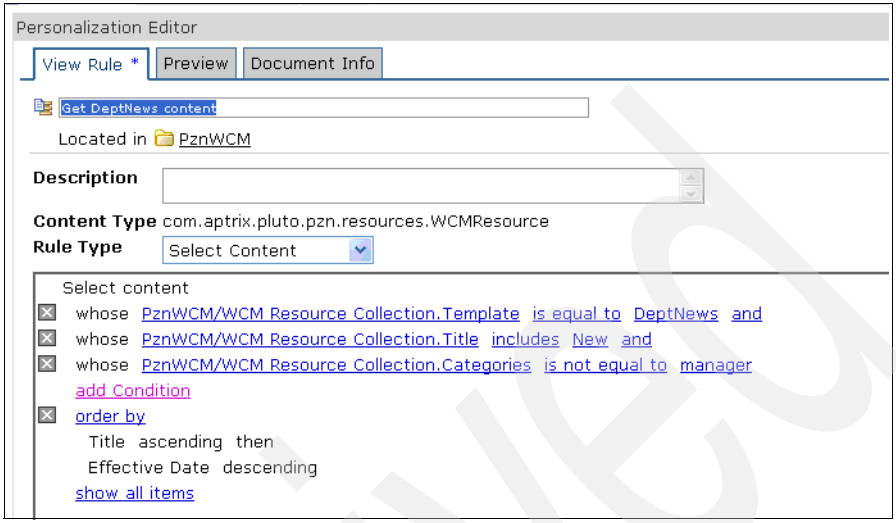


Figure 11-16 Excluding the manager items

4. Tie this together by creating a binding rule that displays the Get DeptNews for both managers and non-managers and that displays manager content for managers (as shown in Figure 11-17 on page 460). A binding rule can use multiple profile names to deliver the targeted content to those users conditionally.

Archived

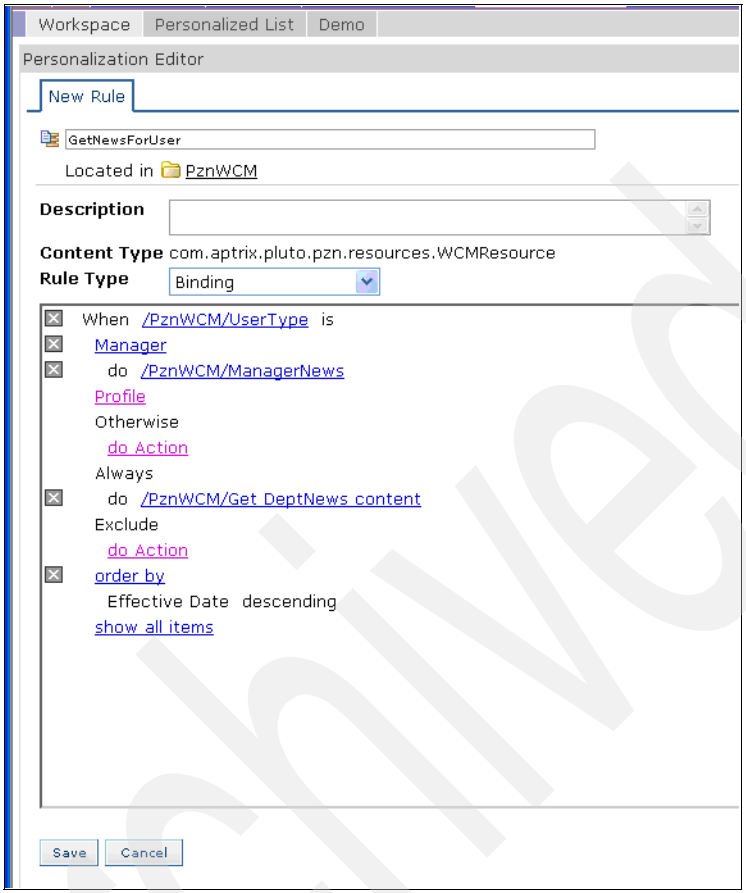


Figure 11-17 Creating a binding rule

Now all the rules are in place, and you can create some content to ensure that mangers and non-mangers see different news items.

Archived

11.3.8 Creating IBM Workplace Web Content Management content using a profile

News items can have a profile that is associated with that item. Profile assignments are selected from a list of taxonomy categories. These categories are created easily using the IBM Workplace Web Content Management authoring user interface (as shown Figure 11-18).

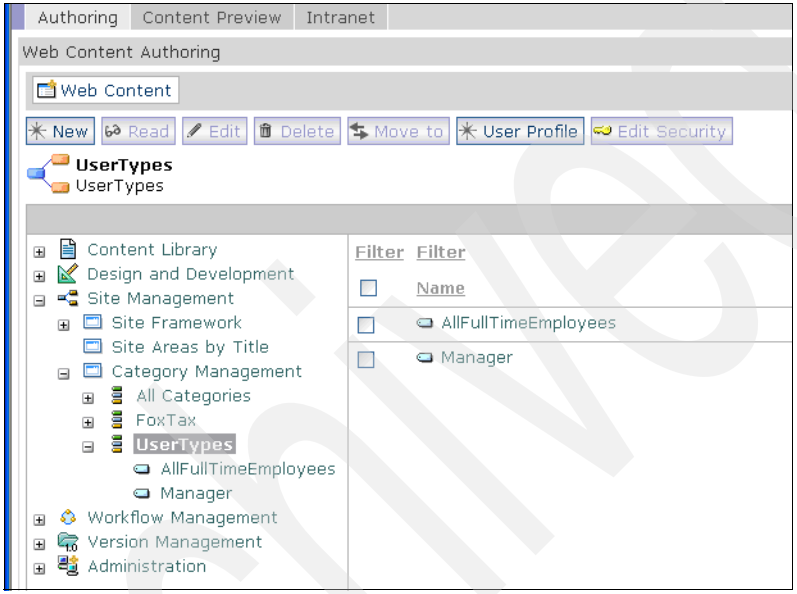


Figure 11-18 Authoring user interface

Archived

For our scenario, we have two categories under the UserTypes taxonomy: AllFullTimeEmployees and Manager. After you have created the categories, you can create a new DeptNews item and select Manager for the Profile, as shown in Figure 11-19.

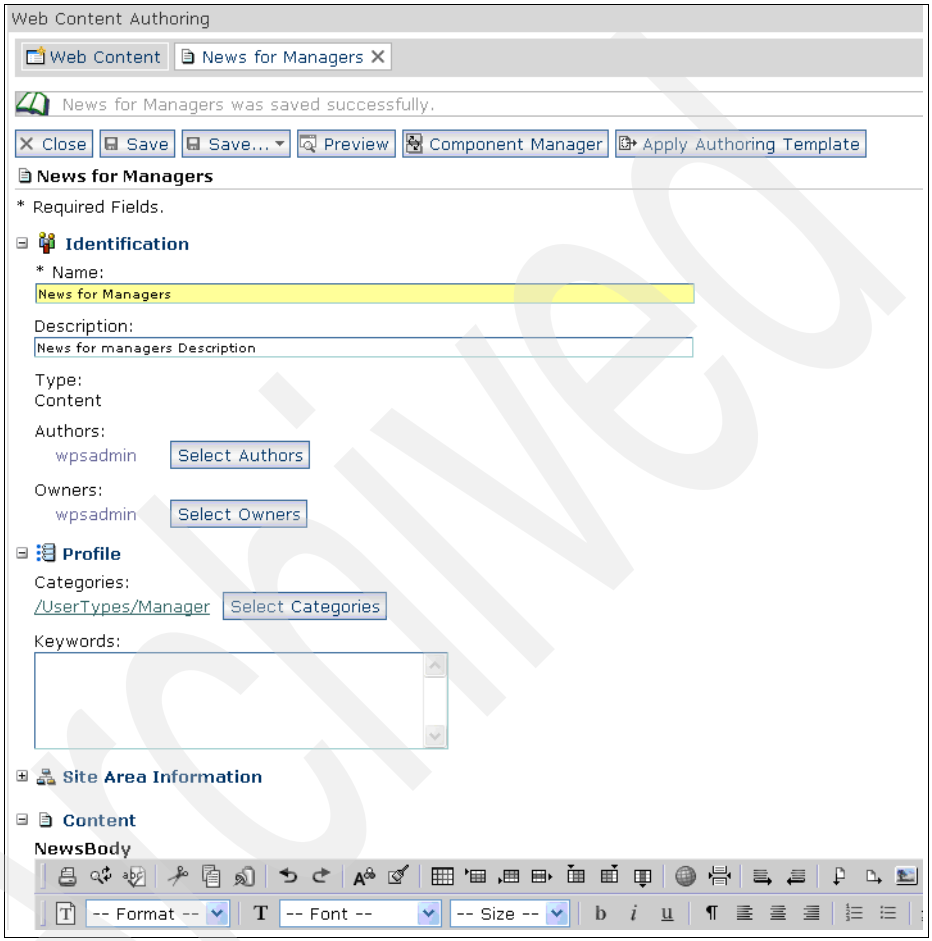


Figure 11-19 Creating a new item

Archived

Next, change the DeptNewsList Personalization component that you created previously to use the new Binding Rule, as shown in Figure 11-20.

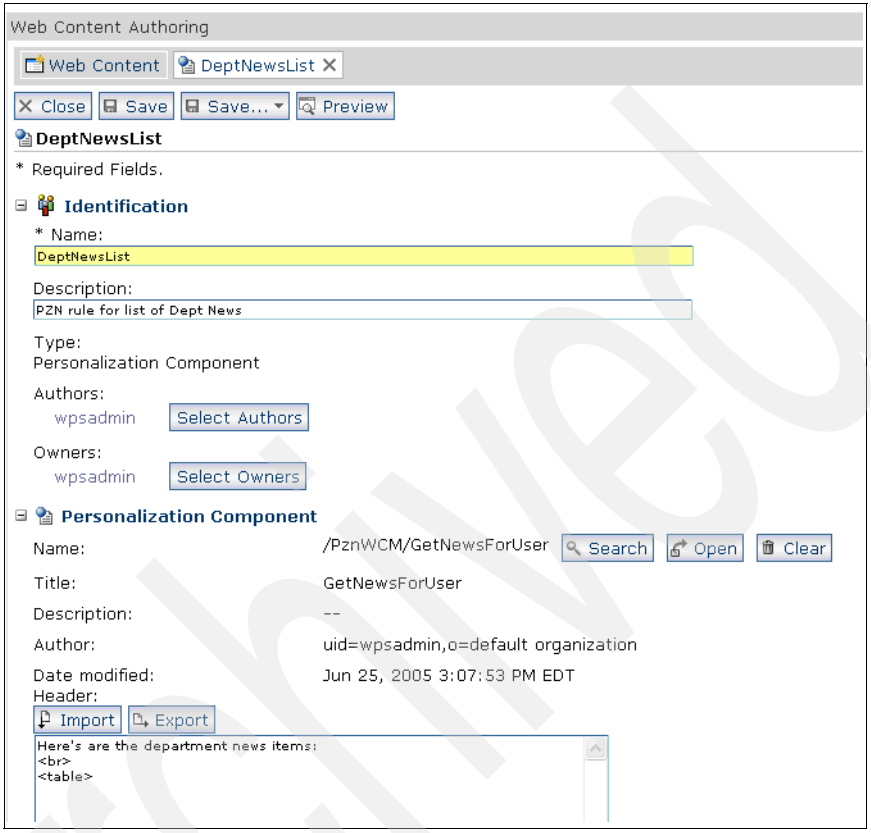


Figure 11-20 Using the new binding rule

We you have the rules and the content ready, next you need to add a Portlet to a page and configure the Portlet to the new presentation page.

Archived

11.3.9 Configure the Portlet in personalization demo

The personalization demo is a great example that comes with WebSphere Portal 5.1. (See the Information Center for instructions on how to install these portlets.) To configure the Portlet:

1. On the personalization demo page, add the content viewer Portlet, as shown in Figure 11-21.

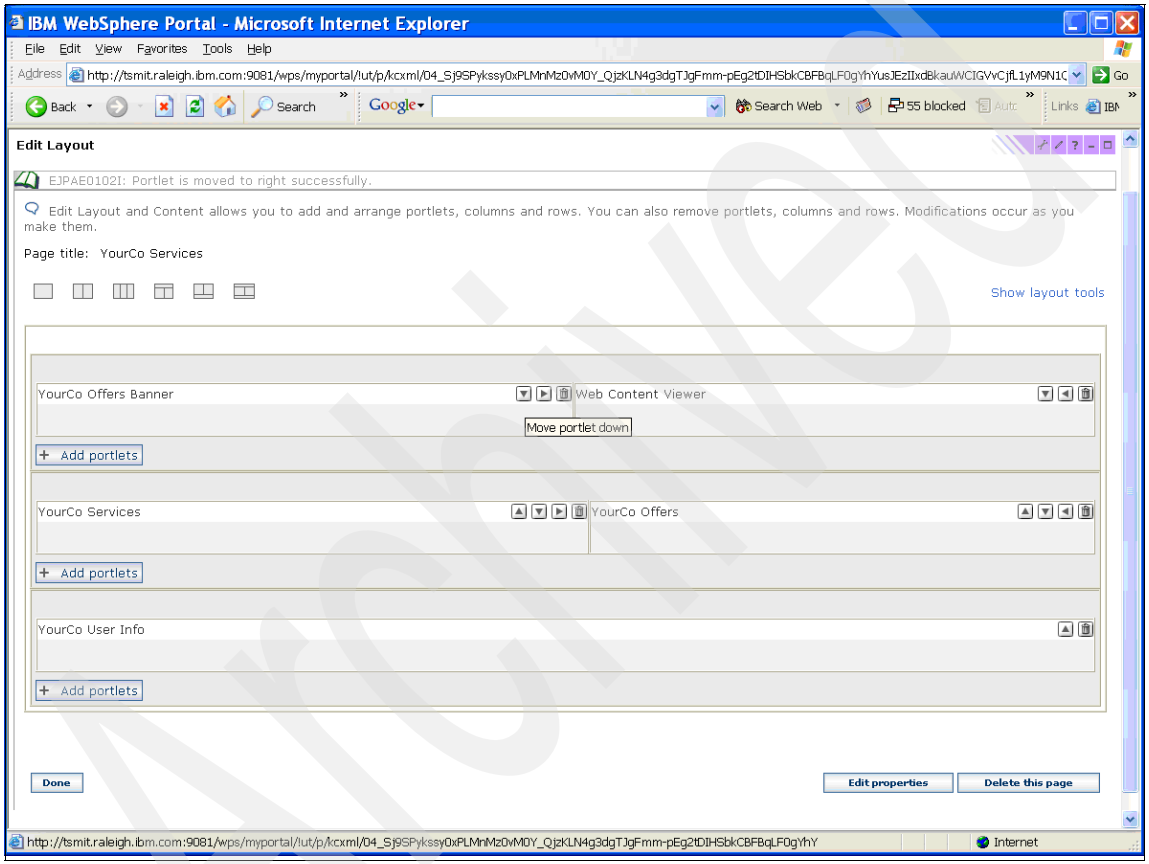


Figure 11-21 Adding the content viewer Portlet

Archived

2. Configure the Portlet to map to the PageDesign that you created that has the personalized list and detail, as shown in Figure 11-22

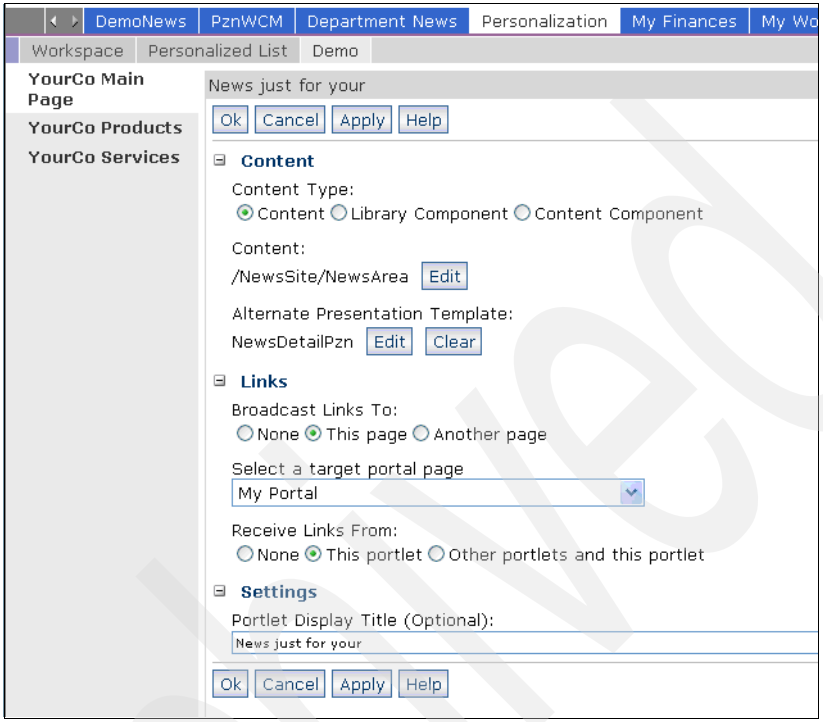


Figure 11-22 Mapping to the PageDesign

Archived

3. Test the new rules. There are a number of IDs that are created by the demo installation task, as shown in Figure 11-23. For example, Scott is not a manager, and Marge is a manager.

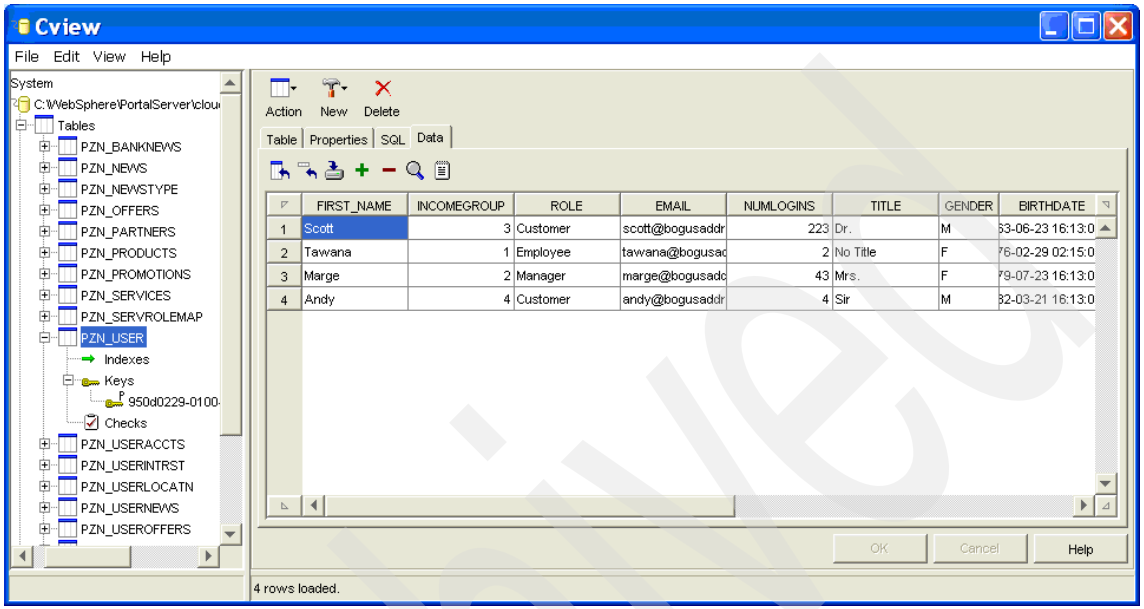


Figure 11-23 IDs created by the demo installation task

Archived

Log in as Scott and navigate to the Personalization Demo. You should see all non-manager news content.

Log in as Marge, and you see your additional item for managers (Figure 11-24).

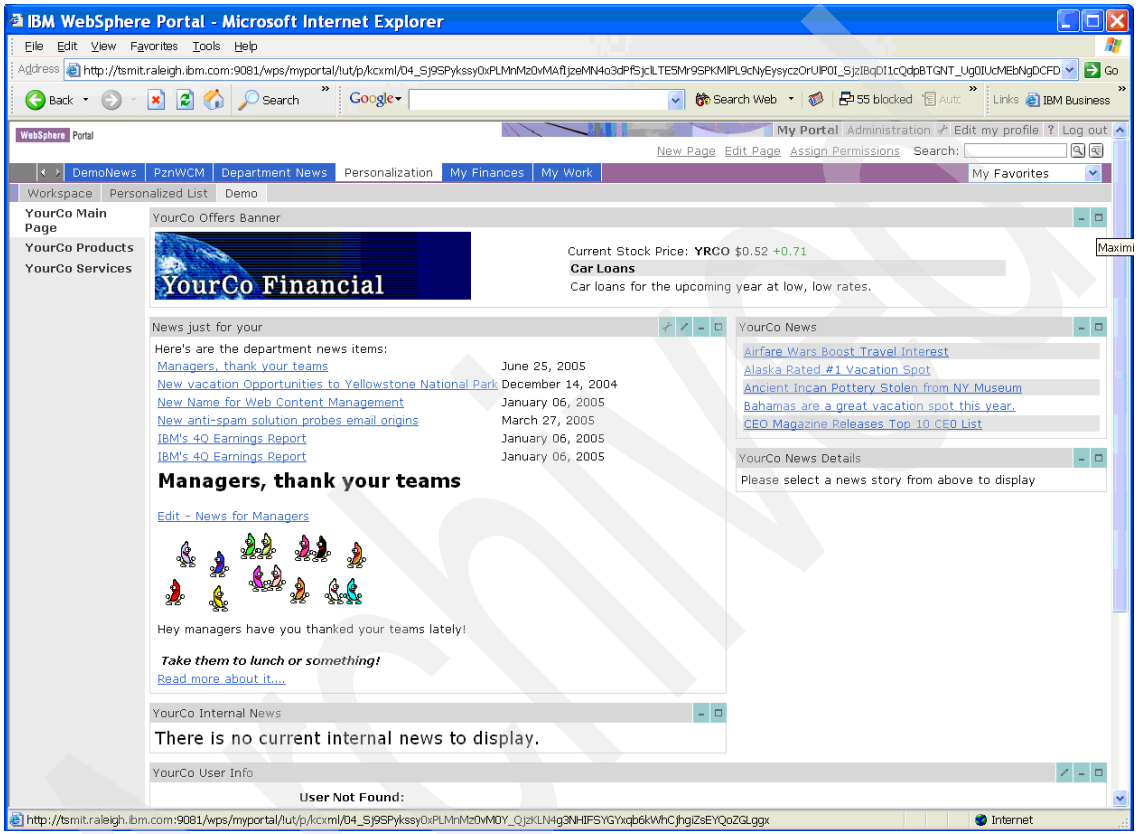


Figure 11-24 Content visible to managers

11.3.10 Including the rule in JSP

Yet another way to use this rule is to include it in a JSP by including a content spot. The Personalization Demo has many examples of rules in a JSP. There are two ways to code a content spot: *explicitly* generated or *dynamically* generated. The rule that is associated with a the spot is specified in the Personalization Workspace, as shown in Figure 11-25 on page 468.

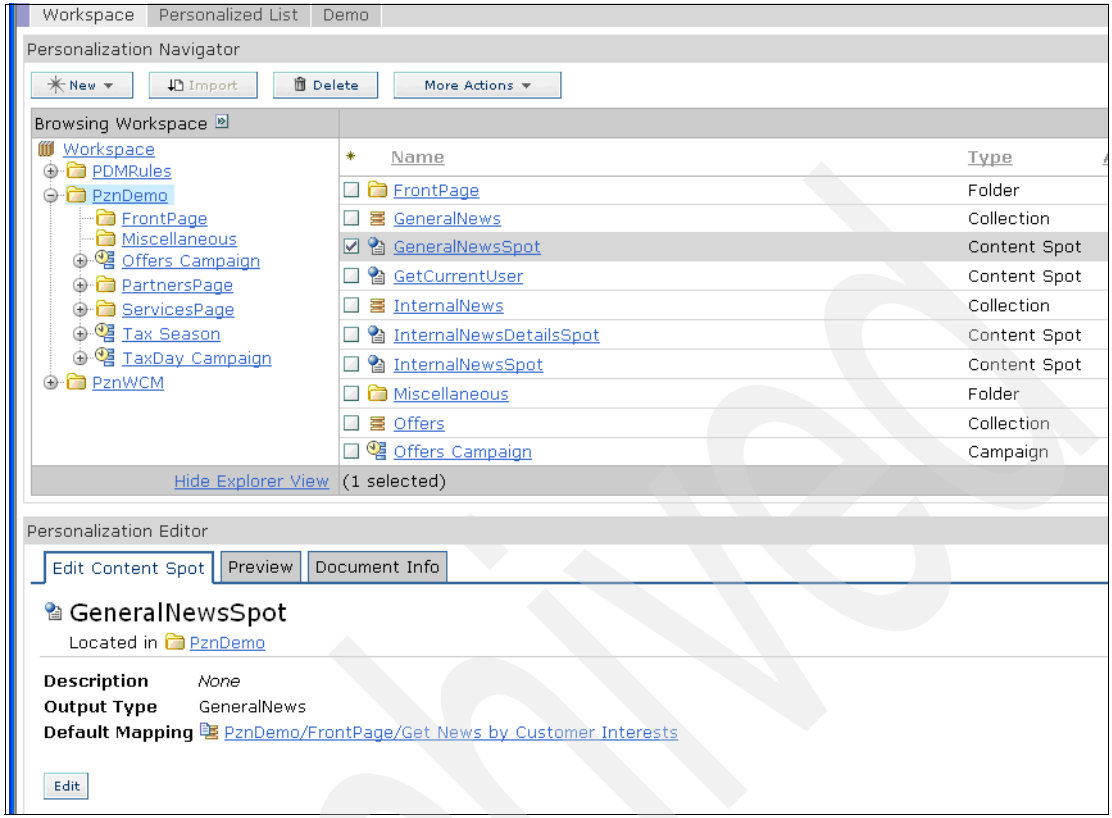


Figure 11-25 The Personalization Workspace

The JSP code remains static. Business users can vary the rules by creating a new rule and tying that rule to the spot, without a developer changing the site.

Archived

12

Working with the application programming interface

This chapter provides details on the IBM Workplace Web Content Management API. It provides you with the ability to create custom Portlets and JavaServer Pages using IBM Workplace Web Content Management functionality.

This chapter includes the following sections:

- Introduction to the API
- Components of the IBM Workplace Web Content Management API
- Extending your Web site with JSP support
- Setting up the development environment
- Examples of API development
- Deployment considerations

© Copyright IBM Corp. 2006. All rights reserved.

12.1 Introduction to the API

The IBM Workplace Web Content Management Java application program interface (API) allows you to access Web content management functionality from custom code and applications. With the IBM Workplace Web Content Management API, you can create custom portlets, servlets, and JSP pages that use IBM Workplace Web Content Management functionality. It provides access to important items during the content creation and rendering process, including content, sites, site areas, taxonomies and categories, and workflows and library components. The IBM Workplace Web Content Management API provides a range of search iterators to access these items. For example, you can find items by their item ID or by a given name and type. The rendering process is achieved through a workspace object which is the entry point into the repository for a particular user.

The current version of the IBM Workplace Web Content Management API does not allow read and write access to all items of the Web content management system. Some functionality is excluded to avoid potential damage to the data integrity or to avoid exposing functionality which can be changed significantly in future releases of the product.

Restriction: You cannot create, update, or delete workflows, workflow stages, workflow actions, taxonomies, categories, syndicators, subscribers, presentation templates, authoring templates, and library components with the IBM Workplace Web Content Management API (with the exception of file resource, HTML, and image library components).

12.1.1 Use cases for the API

In addition to the integration of Web content management functionality within custom portlets, you can use the IBM Workplace Web Content Management API in multiple scenarios as shown in the following examples:

- Bulk resource import

  You can use the IBM Workplace Web Content Management API to create a bulk resource import of a large number of files or images into IBM Workplace Web Content Management.

- Import site form HTML

  After some initial preparation work in IBM Workplace Web Content Management, you can use the IBM Workplace Web Content Management API to import a flat HTML site. The development of the import process and the preparation work can be as complex as necessary. For example, the

importer could break the pages into different components or could reference library components.

- Import site from third-party Web content management system

  Similar to the import of a flat HTML site, you can use the IBM Workplace Web Content Management API to migrate an existing Web site from a third-party Web content management system. In addition to the raw content, this scenario also includes the migration of additional metadata such as authors, keywords, or descriptions. You can also migrate other content stores (for example, Lotus Domino databases).

- Custom site export site to HTML or XML

  Similar to the pre-rendering (`Cacher`) functionality of IBM Workplace Web Content Management, you can use the IBM Workplace Web Content Management API to render an entire site or subset of pages. The output can be generated in any format, such as HTML or XML, to perform custom rendering transformation.

- External workflow applications

  A more complex example for the usage of the IBM Workplace Web Content Management API is the extension of the built-in workflow capabilities through an external workflow application. The external workflow application can push documents based on rules around defined workflow stages. On the other side, external applications can poll documents from their workflow stage and perform some action before moving documents to the next stage.

12.2 Components of the IBM Workplace Web Content Management API

This section describes the core class and interfaces of the IBM Workplace Web Content Management API. It introduces the basic objects and components of IBM Workplace Web Content Management and then connects these objects and components with the corresponding application programming interfaces.

When you use IBM Workplace Web Content Management to develop a Web site, you act as a registered user. All activities that you perform are associated with this user. Thus, you must have the authority to perform certain functions as this registered user. In the IBM Workplace Web Content Management API, this authorized access is represented by a *Workspace* object. The Workspace is the heart of the API. Objects are created, saved, deleted, and searched for in the Workspace object. A Workspace is basically an interface to IBM Workplace Web Content Management that is associated with a user. Using a Workspace object, you can perform operations as that authorized user.

Archived

Figure 12-1, Figure 12-2 on page 473, and Figure 12-3 on page 473 show an architectural overview of the IBM Workplace Web Content Management API interface hierarchy.

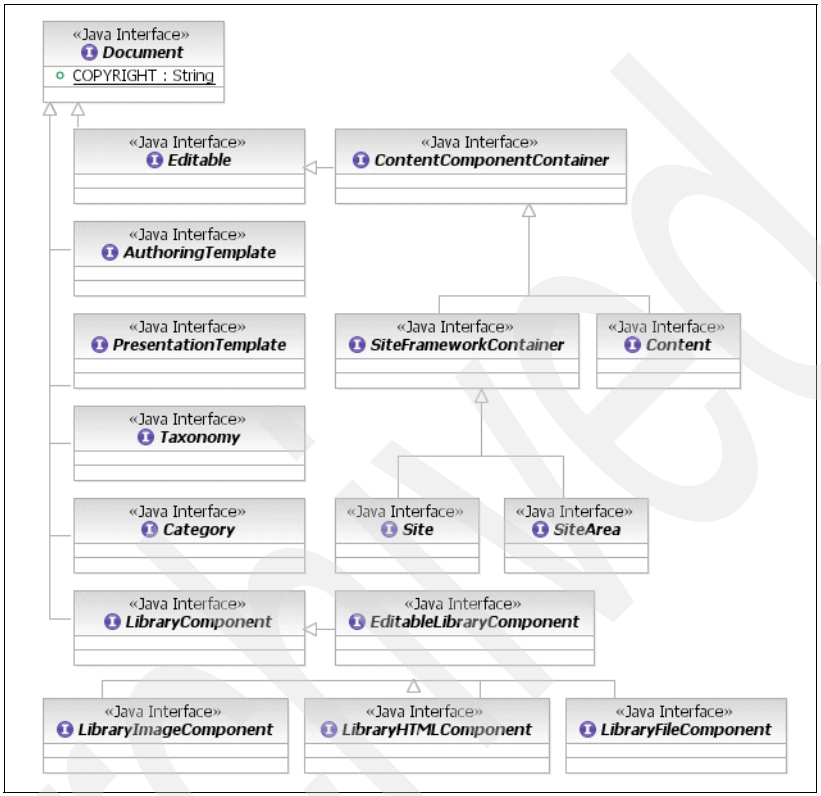


Figure 12-1 Architectural overview , the API interface hierarchy - Part 1

Archived

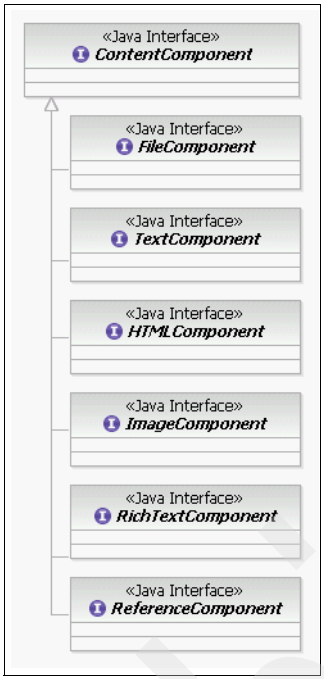


Figure 12-2 Architectural overview, the API interface hierarchy - Part 2

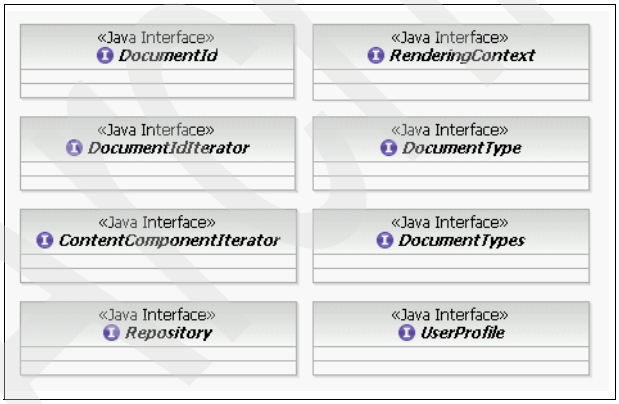


Figure 12-3 Architectural overview, the API interface hierarchy - Part 3

The IBM Workplace Web Content Management API design follows a concept that each interface has only one super-interface. Because of the design decision, the interface Editable and EditAbleLibrary components contain the same methods. The IBM Workplace Web Content Management API does not support native remote access to the IBM Workplace Web Content Management server. It

Archived

only supports Java code that runs in the same Java Virtual Machine as IBM Workplace Web Content Management itself.

> **Note:** Only one class is provided in the IBM Workplace Web Content Management API — the `WCM_API` class. The remaining classes are interfaces.

To read about the complete set of features for the IBM Workplace Web Content Management API, review the JavaDocs that are available via the API. The JavaDocs are located in the following folder on your IBM Web Content Management server:

<*Drive*>:\<*Path*>\WebSphere\AppServer\installedApps\<*Node*>\wcm.ear\ilwwcm.war\webinterface\api-javadoc

You can find further information about the IBM Workplace Web Content Management API in the section *Advanced Features Guide - The Web Content Management API* in the WebSphere Portal Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

12.2.1 The WCM_API class

The `WCM_API` class is the starting point of when you use the IBM Workplace Web Content Management API. This class provides a method to get the repository.

12.2.2 The repository interface

The repository interface provides methods for obtaining a Workspace. A Workspace represents a view of the repository for a specified user. The repository interface provides the following methods to obtain the workspace:

- `getWorkspace(java.security.Principal p_principal)`

  Returns a Workspace object that represents a view of the repository for the specified principal.

- `getWorkspace(java.lang.String p_username, java.lang.String p_password)`

  Returns a Workspace object that represents a view of the repository for a user with the specified credentials.

- `getAnonymousWorkspace()`

  Returns a Workspace object that represents a view of the repository for the Anonymous user.

- `endWorkspace()`

  Closes the current Workspace when the user is finished with it.

The following code shows an example of how to obtain a repository instance from the `WCM_API` class:

```
Repository repository = null;
repository = WCM_API.getRepository();
```

12.2.3 Workspace interface

A Workspace represents a view of the IBM Workplace Web Content Management repository for a particular user. Most of the important functions are provided within the Workspace interface. Operations that are available in the Workspace include:

- Searching for a document.

  The IBM Workplace Web Content Management API provides two kinds of search methods.

  – You can search for documents by providing a detailed search criterion. The criterion includes template ID, siteArea ID, category ID, and keyword.

  – You can search for documents by several provided `findby` methods, including `findbyName`, `findbyType`, `findContentByCategory`, and so on.

- Creating new objects of editable types, such as content, site, component, and so on.
- Rendering given context or rendering a given component using specified context.
- Saving and deleting editable objects.

To get a Workspace object, you must request one from the repository singleton. The Workspace object is the entry point into the IBM Workplace Web Content Management repository. For example, to get a Workspace object for a registered user, use the following call:

```
WCM_API.getRepository().getWorkspace(username, password)
```

Note: If the user is not recognized as an IBM Workplace Web Content Management user or if the user cannot be authenticated, an `OperationFailedException` is thrown.

Archived

Within the portal context, the current user could by retrieved by the portal request, as shown in Example 12-1.

Example 12-1 Access workspace object for a portal user

```
Workspace ws = null;
Repository repository = null;
try {
   repository = WCM_API.getRepository();
   ws = repository.getWorkspace((java.security.Principal)request.getUser());
} catch (ServiceNotAvailableException e) {
   e.printStackTrace();
} catch (OperationFailedException e) {
   e.printStackTrace();
}
```

12.2.4 Document object

A document represents an object or an entity in the repository. A document is the parent interface of most other entity interfaces, such as content, category, site, and so on. In most cases, you only use a document's subinterfaces. The interface contains methods to access general attributes of the repository entity, as follows.

- `getName()`

  Returns the name of the document object

- `getID()`

  Returns the document ID of the document object

- `getDescription()`

  Returns the description of the document object.

The document interface also contains several methods to retrieve the members of different security level, as follows:

- `getAuthors()`
- `getDeleteAccessMembers()`
- `getEditAccessMembers()`
- `getLiveAccessMembers()`
- `getOwners()`
- `getReadAccessMembers()`

Each method returns a string array that contains the members. If the document does not have the requested access right specified, the methods return an empty array.

Archived

Note: The created documents do not exist in the repository until they are saved.

12.2.5 DocumentID interface

Each entity in the repository has a unique identifier, called *DocumentID*, that uniquely represents this entity. Different entities can have the same name, but they must have different IDs. The DocumentID is represented in the IBM Workplace Web Content Management API by the DocumentID interface. This interface contains the following methods:

- `getName()`

  Returns the name of the document with this DocumentID.

- `getType()`

  Returns the type of the document with this DocumentID.

- `isOfType(com.ibm.workplace.wcm.api.DocumentType p_type)`

  Returns `true` if, and only if, the type of the document that is represented by this DocumentID is the same as the `DocumentType` argument.

- `toString()`

  Returns a string representation of this DocumentID.

Attention: The DocumentID format is not guaranteed to be consistent between releases. Do not store the string representation and use later.

12.2.6 DocumentType and DocumentTypes interfaces

A *document type* represents a kind of document. The document type can be content, category, site, or any of the components. In the IBM Workplace Web Content Management API, the *DocumentType* and *DocumentTypes* interfaces handle document types. The DocumentTypes interface maintains a list of all available document types, and a DocumentType object is an individual document type.

The DocumentType interface contains the `toString()` method, which returns a string representation of the objects class name. The string values are stored as constants in the DocumentTypes interface.

Archived

12.2.7 RenderingContext interface

In IBM Workplace Web Content Management, it is a basic requirement to render content or components in a Web page. Due to the dynamic presentation feature when rendering components, you need to get the context for rendering first. The RenderingContext interface represents the context of rendering that contains information that is required to present a renderable object. The contained information includes the current content, page design and site, the servlet path and the application path, the rendered content, and so on.

The source code fragment shown in Example 12-2 includes all the steps that are necessary to render a content page. This example assumes the existence of a request and response such as within a servlet or JSP.

Example 12-2 RenderingContent method

```
Workspace workspace = null;
RenderingContext context = null;
workspace = WCM_API.getRepository().getWorkspace("user", "password");
HashMap map = new HashMap();
context = workspace.createRenderingContext(request, response, map);
context.setRenderedContent("/Intranet/IT Home/News and Events/News/");
String content = workspace.render(context);
```

12.2.8 Editable components of the interfaces

This section discusses the editable components of the interfaces. In IBM Workplace Web Content Management, there are two kinds of editable components: *content components* and *library components*.

Content components are components that are stored in content, site, or site area components. *Library components* are static components that are stored directly in the component library.

Content components and library components are both retrieved by the page design. However, when you retrieve library components in a page design, the components that are retrieved are known at all times because they are directly referenced. As for content components, the actual component that is rendered is dependent upon the current content or site, because they are referenced dynamically when content is rendered.

ContentCompontentContainer interface

The ContentCompontentContainer interface is the root interface for the content component container hierarchy. A ContentComponentContainer represents a container for ContentComponent objects.

The ContentComponentContainer interface includes methods for retrieving and setting ContentComponent objects. ContentComponents can be added to site, sitearea, and content objects. The ContentComponentContainer interface provides basic functions such as the updating and retrieving of content components from their container.

The following list shows examples for methods that are included within the ContentComponentContainer interface:

- `componentIterator()`

  Returns an iterator of all ContentComponent objects in this ContentComponentContainer.

- `getComponent(java.lang.String p_name)`

  Retrieves a ContentComponent with a name specified by the argument from this ContentComponentContainer.

In the hierarchy that is below the ContentComponentContainer interface, the API implements interfaces for the content and the site framework.

Content objects

Content objects store values that can be referenced in other IBM Workplace Web Content Management objects. These values include ID information, profiling, security, and components. Content is represented by the content interface. You can create and save content objects through the API. You can also retrieve existing content object to be edit and save them. The content interface provides methods to access the items workflow properties, including moving the item to the next stage in the workflow, creating a draft, checking its published or expired status, and so forth. The following list shows a few examples for most common methods of the content interface:

- `getAuthoringTemplateID()`

  Returns the ID of the AuthoringTemplate that was used to create this Content document.

- `nextWorkflowStage()`

  Move this Content object into the next stage in its workflow.

- `getWorkflowId()`

  Returns the ID of the workflow associated with this content.

- `setKeywords(java.lang.String[ ] p_keywords)`

  Sets the keywords for this Content document.

Refer to the API documentation for a complete list of methods for the content interface.

Archived

SiteFrameworkContainer interface

The SiteFrameworkContainer interface implements the ability to organize and build a site framework. A site framework is a hierarchy consisting of site, site area, and content objects. The site and site area extends the SiteFrameworkContainer interface to provide a uniform interface when organizing and building the site framework.

The following list shows a few examples for most common methods of the SiteFrameworkContainer interface:

- `setDefaultContent (...) / getDefaultContent()`

  Sets or gets the DocumentID of the default content for this site or site area.

- `addTemplateMapping(...) / getTemplateMapping(...)`

  Adds or gets a mapping for the specified authoring template and presentation template.

Refer to the API documentation for a complete list of the methods for this interface.

Example 12-3 iterates through all site area's within all sites and adds names to an ArrayList.

Example 12-3 SiteArea ArrayList

```
DocumentIdIterator siteIterator = ws.findByType(DocumentTypes.Site);
Site siteDocument = null;
ArrayList contentPaths = new ArrayList();
while (siteIterator.hasNext()){
   try {
      siteDocument = (Site)ws.getById((DocumentId) siteIterator.next());
   } catch (DocumentRetrievalException e1) {
      e1.printStackTrace();
   } catch (AuthorizationException e1) {
      e1.printStackTrace();
   }

   DocumentIdIterator siteareaIterator = siteDocument.getChildren();
   SiteArea siteareaDocument = null;
   while (siteareaIterator.hasNext()){
   try {
      siteareaDocument =
(SiteArea)ws.getById((DocumentId)siteareaIterator.next());
      contentPaths.add(siteareaDocument.getName());
      } catch (DocumentRetrievalException e2) {
         e2.printStackTrace();
      } catch (AuthorizationException e2) {
         e2.printStackTrace();
```

```
        }
    }
}
```

LibraryComponent interface

For library components, the IBM Workplace Web Content Management API provides a LibraryComponent interface to represent all the library components. Also, there are LibraryFileComponent, LibraryHTMLComponent, and LibraryImageComponent interfaces as follows:

- LibraryFileComponent

  A LibraryFileComponent can contain one file. The contents of the file are dealt with as raw byte content. This interface provides methods to manipulate the file.

- LibraryHTMLComponent

  A LibraryHTMLComponent can contain HTML. This interface provides methods to retrieve and set the HTML contained within this component.

- LibraryImageComponent

  A LibraryImageComponent can contain one image. The contents of the image file are dealt with as raw byte content. This interface provides methods to manipulate the image file and the attributes to use when displaying the image.

Other library components, such as the navigator and menu components, do not have corresponding interfaces in the API.

ContentComponent interface

The ContentComponent interface represents the root interface in the content component hierarchy. ContentComponent marks a component that can be held within a ContentComponentContainer. A ContentComponent can exist only on content, site or site area objects. A ContentComponent must live in a ContentComponentContainer. It cannot exist in the repository as a separate object.

A content component has a one-to-one relationship with a ContentComponentContainer object, and consequently, with an object that can contain a ContentComponentContainer.

The ContentComponent interface includes the following methods:

- `getContainer()`

  This method returns the ContentComponentContainer object that contains this ContentComponent.

Archived

- `getName`

  Returns the name of this ContentComponent object.

The interface acts as super-interface for the content components. The IBM Workplace Web Content Management API provides the following content components:

- File
- HTML
- Image
- Reference
- RichText
- Text

Like the ContentComponent, the content components cannot be stored as a separate entity in the repository.

Each of the subinterfaces contains different `get` and `set` methods to manipulate the data that is contained within this component. For example the TextComponent interface contains the following methods:

- `getText()`
- `setText()`

Refer to the IBM Workplace Web Content Management API JavaDocs for a complete reference of the subinterface methods.

> **Note:** The JSPComponent and Federated content components are not provided through the IBM Workplace Web Content Management API.

12.2.9 Read-only component interfaces

In addition to the components that you can edit, the IBM Workplace Web Content Management API contains read-only interfaces that you can use access to certain components. This sections describes those component interfaces that you cannot edit.

> **Note:** You cannot create, edit, or delete the objects that this section discusses through the IBM Workplace Web Content Management API. They must already exist in the repository. You must create them manually through the authoring portlet.

Archived

AuthoringTemplate

The AuthoringTemplate subinterface represents an authoring template. This template is the basis of a content object. A new content object derives its default attributes from this template. The AuthoringTemplate subinterface inherits methods form the Document interface and does not contain its own methods.

PresentationTemplate

A PresentationTemplate is a template for how a Web page or the content in the Web Content Viewer portlet looks when rendered. Like the AuthoringTemplate, the PresentationTemplate inherits its methods from the Document interface and does not implement its own methods.

Category

The Category interface provides methods that aid in traversing a category tree. The interface extends the Document interface and includes the following additional methods:

- `getParent()`

  Returns the DocumentID of the parent of this Category object.

- `getChilden()`

  Retrieves the immediate children of the Category object.

- `getAllChildren()`

  Returns an depth first iterator of DocumentID objects of the immediate child categories of this Category object.

For example, let us assume that our category tree contains the structure shown in Figure 12-4.

```
MyCategory
   CategoryA
      CategoryA1
         CategoryA1-1
         CategoryA1-2
      CategoryA2
   CategoryB
      CategoryB1
```

Figure 12-4 Sample category tree

Archived

If we make a call to the `getAllChildren()` method of the object that represents `MyCategory`, we get an iterator that contains the following categories:

- CategoryA
- CategoryA1
- CategoryA1-1
- CategoryA1-2
- CategoryA2
- CategoryB
- CategoryB1

A call of the `getChildren()` method of the same object returns an iterator that contains categories CategoryA and CategoryB.

> **Attention:** The returned iterator only contains objects that have a least read access for the current user.

Taxonomy

The Taxonomy interface is very similar to the Category interface. The only exception is that a taxonomy object does *not* have an ancestor. Therefore, the interface does not contain a `getParent()` method.

Workflow

The Workflow interface represents a Workflow within in repository. A Workflow consists of WorkflowStages in a specific order. The Workflow controls access to and manages the progression of IBM Workplace Web Content Management objects through the specified WorkflowStages. The interface contains the following methods to iterate through the Workflow stages:

- `getStagesIterator()`

  Returns an iterator of DocumentID objects of the WorkflowStages in this Workflow.

- `getNumStages()`

  Returns the number of stages in this Workflow.

WorkflowStage

A WorkflowStage represents a step of a Workflow definition. The interface inherits its methods from the Document interface and does not implement its own methods.

Archived

12.2.10 UserProfile

The UserProfile interface allows access to user related information for a user who is logged in and validated. The interface provides the following methods:

- `getCategories()`

  Returns a string array of the categories associates with this user.

- `getKeywords()`

  Returns a string array of the keywords associated with this user.

- `getUsername()`

  Returns the username of the user in this set of credentials.

12.2.11 Iterator interfaces

The IBM Workplace Web Content Management API provides the following interfaces to iterate through documents:

- DocumentIdIterator
- ContentComponentIterator

Both interfaces extend the java.util.Iterator interface and contain the following methods:

- `hasNext()`

  Returns `true` if the iteration has more elements.

- `next()`

  Returns the next element in the iteration. All objects returned from this iterator are instances of DocumentID.

> **Attention:** The `remove()` method of the java.util.Iterator interface is not implemented in both interfaces and throws an `UnsupportedOperationException`.

12.3 Extending your Web site with JSP support

If you are planning to install IBM Workplace Web Content Management, chances are that you have existing Web resources (such as applications or business logic) that you want to integrate into your Web site. IBM Workplace Web Content Management provides a number of new features for developers to extend the capabilities of their Web content management implementations. This release includes the support for JavaServer Page (JSP) technology. JSPs enable you to

Archived

develop dynamic Web pages that you can integrate with existing business logic by using embedded Java code in HTML files. IBM Workplace Web Content Management provides JSP technology through the JSP component and a JSP tag library.

Designing for Portlets

When designing your IBM Workplace Web Content Management portlet applications, you generally use the MVC Model 2 architecture pattern. Like the servlet MVC implementation, the view is traditionally implemented using JSPs.

The JSP uses custom tags, which are user-definded JSP language elements that encapsulate reusable java code. JSP tags are used to access components and content from IBM Workplace Web Content Management content easily. You can use the IBM Workplace Web Content Management JSP tag library in stand-alone JSP pages, which also includes JSP used within a portlet life cycle and IBM Workplace Web Content Management JSP components.

12.3.1 JSP components

The JSP component references a JSP from within the IBM Workplace Web Content Management framework. This reference provides site developers with a number of capabilities. For example, you can use this new JSP component to integrate existing JSPs into a IBM Workplace Web Content Management Web site. Therefore, if you have existing JSPs that are used for generating dynamic content or business logic, you can integrate these JSPs into IBM Workplace Web Content Management. In addition, the JSP can serve as a container for additional business logic or can reference a servlet or bean. Thus, you can use the JSP to contain IBM Workplace Web Content Management custom extensions, such as integration with other applications (backend data, transactional systems, or even Lotus Domino).

Note: All JSPs referenced using the new JSP component need to reside within the IBM Workplace Web Content Management application. Therefore, the JSP file needs to reside under the Web-app directory of the WebSphere Application Server on which the JSP is running.

Note: The JSP must be stored within the following directory of your IBM Workplace Web Content Management server:

../AppServer/installedApps/[node-name]/wcm.ear/ilwwcm.war/jsp

You might also need to store the JSP in the client war directory of the servlet or portlet that calls the JSP if you are using the IBM Workplace Web Content Management API.

Note: Syndication does not move the JSP that is referred to in a JSP component. Only the component itself is moved. You need to store a JSP on both the subscribing *and* the syndicating servers.

12.3.2 JSP tag library

In addition to the JSP component, IBM Workplace Web Content Management includes a JSP tag library that encapsulates the core functionality common to the IBM Workplace Web Content Management API. IBM Workplace Web Content Management ships with a library of JSP tags that you can use to display IBM Workplace Web Content Management components and content from an external JSP. These tags provide new menus in existing JSPs and applications, navigational components, or content. As with the JSP component, all JSPs must reside within the IBM Workplace Web Content Management application.

The JSP tag library is named wcm.tld and contains the following tags:

- `initworkspace`
- `setExplicitContext`
- `setContext`
- `content`
- `contentComponent`
- `libraryComponent`

The JSP tag library is located in the \wcm.ear\ilwwcm.war\WEB-INF\tld directory below the installedApps directory of the WebSphere Application Server. You can reference the library as shown in the following example:

```
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
```

The tag library is not included in the Portlet projects that are created using the RAD Portlet Tools. For this reason, you need to add the wcm.tld file to portlet projects manually to avoid broken link warning messages during the development of JSP.

The initworkspace tag

The initworkspace tag is used to set the initial workspace. Each workspace is associated which a specified user. Therefore, the initworkspace tag contains parameters that provides a user name and password, as shown in Example 12-4.

Example 12-4 JSP tag library, initworkspace tag

```
<wcm:initworkspace username="<username>" password="<passord>" >
Could not init workplace
</wcm:initworkspace>
```

If you use the JSP tag library in the WebSphere Portal context, you should use the `getUser()` property of the PortalRequest to get the current user name, as shown in Example 12-5.

Example 12-5 The initworkspace tag retrieves the user name from the portalRequest

```
<wcm:initworkspace user="<%= (java.security.Principal)portletRequest.getUser()
%>" />
```

The setExplicitContext tag

After you have initialized the IBM Workplace Web Content Management workplace, you then need to use the explicit context tag, setExplicitContext. This tag tells the JSP and application server the location of your IBM Workplace Web Content Management instance, as shown in Example 12-6.

Example 12-6 setExplicitContext tag syntax

```
<wcm:setExplicitContext wcmWebAppPath=" " wcmServletPath=" " path=" " >
[Error Message]
</wcm:setExplicitContext>
```

For our example, the JSP and IBM Workplace Web Content Management are located on the same server, because currently, the tag library is implemented for local Web container access to Web content management components. Therefore, our settings are as shown in Example 12-7.

Example 12-7 setExplicitContext within a stand-alone JSP

```
<wcm:setExplicitContext wcmWebAppPath="http://localhost:9080/content"
wcmServletPath="/connect" path="/Intranet">Error in explicit path to WCM
server</wcm:setExplicitContext>
```

When we use the JSP tag as a View component within a Portlet, it usually does not make sense to the use static path information. Instead of static information, you should use Portal constancies, as shown in Example 12-8.

Example 12-8 setExplicitContent within WebSphere Portal context

```
<% String appPath =
"http://"+portletRequest.getServerName()+":"+portletRequest.getServerPort()+"/w
ps/wcm"; %>

<wcm:initworkspace user="<%= (java.security.Principal)portletRequest.getUser()
%>" />
<wcm:setExplicitContext wcmServletPath="/connect" wcmWebAppPath="<%=appPath%>"
path="/RiverBend/Company">
```

Notice that Example 12-8 includes the `wcmWebAppPath` and `wcmServletPath` parameters. These parameters are optional. They are used to specify the servlet and context root of the IBM Workplace Web Content Management Web site. These parameters are important for the rewriting of URLs when the JSPs are located in a path that is different from the Web content. If you include `wcmWebAppPath`, you must also specify `wcmServletPath`.

The setContext tag

The setContext tag is very similar to the setExplicitContext tag. The setExplicitContext tag takes the content path as one of its parameters, while the setContext tag looks for the content path in the specified location (query, request, or session) with the specified key. The content path value that is either specified in the tag or retrieved from the request attributes or parameters or session is the same for either tag, as shown in the following example:

```
<wcm:setContext location=" " wcmWebAppPath="" wcmServletPath="" param=" " >
```

The content tag

The Content tag renders the content based on the current context. An optional parameter is provided to set a specified page design. The default page design that is associated with the content is used if the page design parameter is not set.

The following example renders the current content by using its default page design:

```
<wcm:content pageDesign="" >
```

The contentComponent tag

The contentComponent tag is similar to the AptrixCmpnt tag. The tag renders a Component from the current site, site area, or content object. The tag includes the two parameter types and names. Both parameters are equal to the AptrixCmpnt tag. Thus, you can provide a specified field as key and content, site area, or site as value for the type as shown in the following example:

Archived

```
<wcm:contentComponent key="Summary" type="content" >
```

The tag shown in this example renders the `Summary` content component of the current content document into the JSP. The context has to be set before. In this example, `Summary` is a text component in the authoring template.

The libraryComponent tag

Like the contentComponent, the libraryComponent tag is very similar to the AptrixLibCmpnt tag. By using this tag, you are able to render different library components into the current JSP. The following example renders the `MENU-Related Information` library component for a previously specified context into the JSP:

```
<wcm:libraryComponent name="MENU-Related Information">
```

Note: You can find a complete description of the tag parameters for AptrixCmpnt and AptrixLibCmpnt in the IBM Workplace Web Content Management Information Center at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

12.3.3 JSP tag library example

Now that we have reviewed the core tags for the library, the next step is to put a JSP together to test the functionality. We have a JSP directory within the Web application folder, so we can use that to store JSP files. Example 12-9 displays the Latest News menu component from the IBM Workplace Web Content Management server.

Example 12-9 Displaying the Latest News menu component

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" %>
<wcm:initworkspace username="Administrator" password="password">
  Could not get WCM workplace</wcm:initworkspace>
<wcm:setExplicitContext wcmWebAppPath=http://localhost:9080/content
wcmServletPath=
  "/connect" path="/Intranet">Error in explicit path to WCM server
  </wcm:setExplicitContext>

<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<TITLE>Latest News</TITLE>
The following is the latest news from the WCM server
</HEAD>
```

```
<BODY LEFTMARGIN=0 MARGINWIDTH=0 TOPMARGIN=0 MARGINHEIGHT=0

<!-- WCM component -->

<wcm:libraryComponent name="MENU-Site Area">
  Could not find Latest News Menu</wcm:libraryComponent>

<!-- End WCM component -->

</BODY>
</HTML>
```

When we save the JSP and run it from the Web browser, the JSP renders the Latest News menu from the IBM Workplace Web Content Management server as shown in Figure 12-5.

Figure 12-5 JSP preview

12.4 Setting up the development environment

This section describes how to install Rational® Application Developer 6 (RAD) and how to attach the development environment to a remote portal server. The remote portal server is already running the IBM Workplace Web Content Management task, as described in 3.2, “Installing IBM Workplace Web Content Management” on page 73.

Note: This section addresses assumes that you are not familiar with Rational Application Developer. If you have experience with portlet development and Rational Application Developer 6, you can skip to 12.4.2, “Installing the API Java files” on page 496.

Archived

12.4.1 Installing Rational Application Developer 6.0

You can install Rational Application Developer 6 (RAD) either from the installation CD-ROM or from a local or network directory. If you want to install from a directory make sure that you copy all files and directories from the first installation CD to a subdirectory called disk1. This subdirectory *must* be called disk1 and it must be in lower case. Create additional subdirectories to copy the contents of the other two CDs to subdirectories called disk2 and disk3. If you install from a network drive, map the drive locally mapped specifying a device name.

To install RAD, follow these steps:

1. Run **`launchpad.exe`** from the root of the first installation CD to display the Rational Application Developer Installation Launchpad window shown in Figure 12-6.

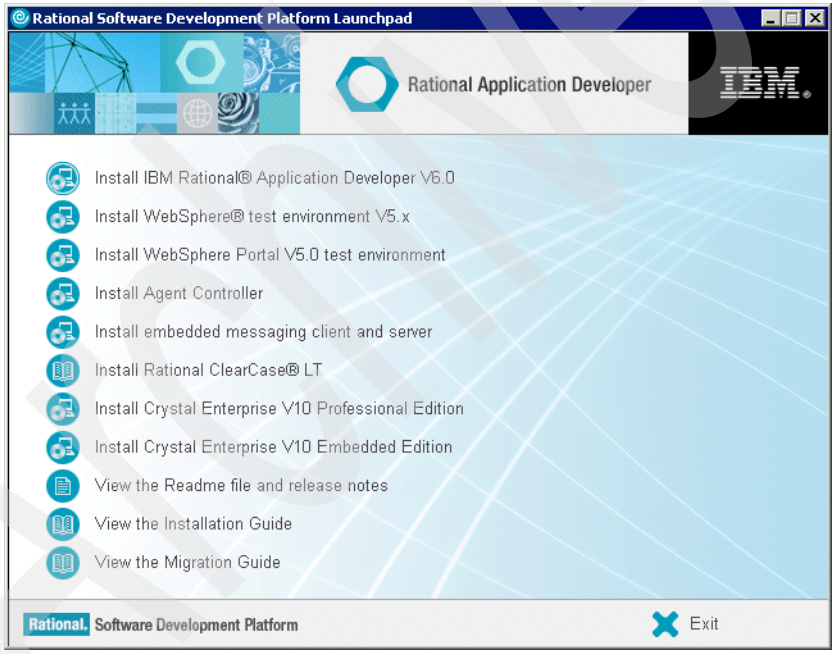


Figure 12-6 RAD Launchpad window

2. From the launchpad, select **Install IBM Rational Application Developer V6.0**.

3. Wait for the installation wizard to open the Welcome window (as shown in Figure 12-7), and then click **Next** to continue the installation.

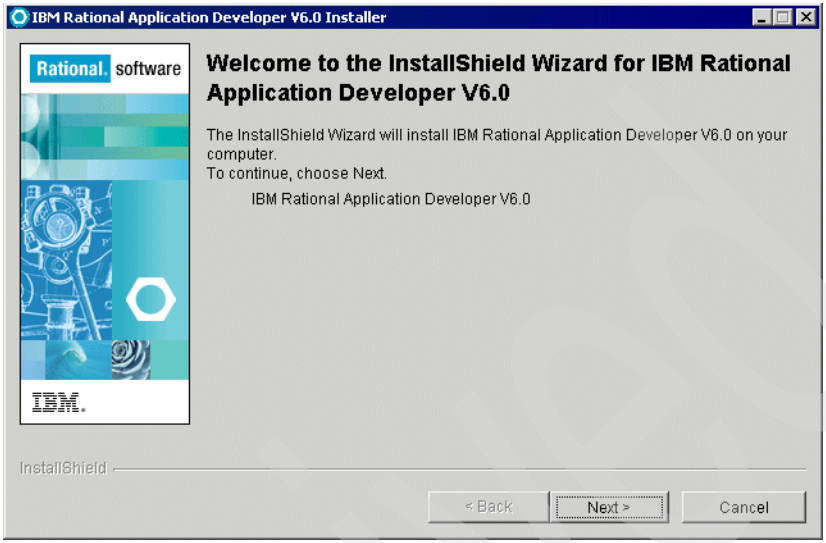


Figure 12-7 RAD Welcome window

4. Follow the instructions for reading and accepting the license agreement (Figure 12-8) and for specifying the target installation directory (Figure 12-9 on page 494).

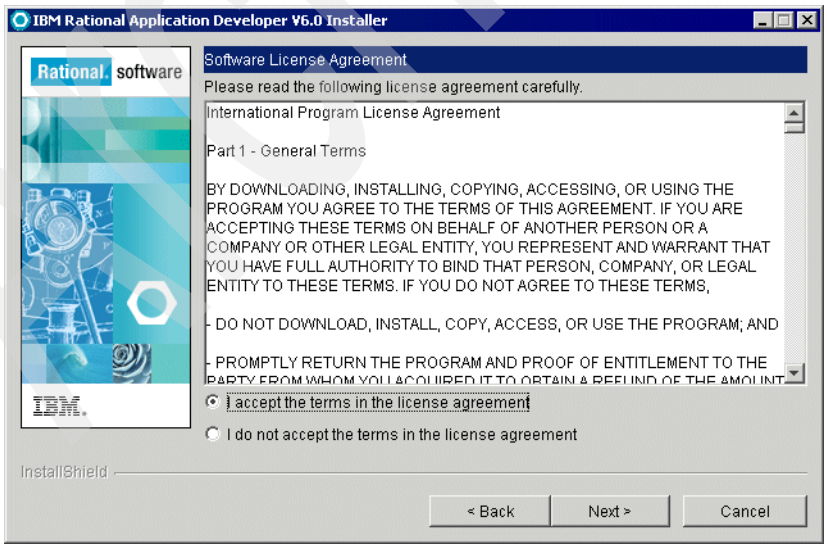


Figure 12-8 RAD License Agreement

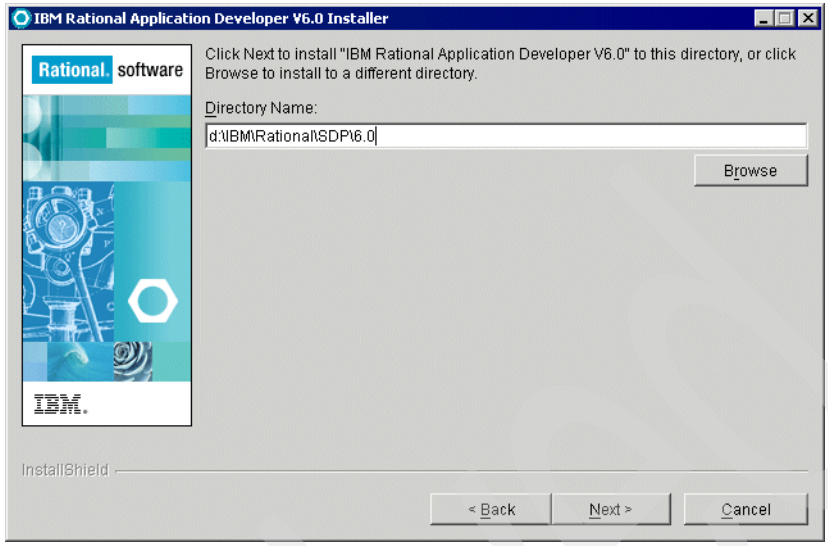


Figure 12-9 RAD target installation directory

5. In the Features window (Figure 12-10 on page 495), select the RAD features that you want to install. By default, the installation program installs the Integrated Development Environment and the optional WebSphere Application Server, Version 6.0 as the run-time environment for testing. If you want to install the user interface and documentation for supported national languages other than English, select to install the Language Pack.

Archived

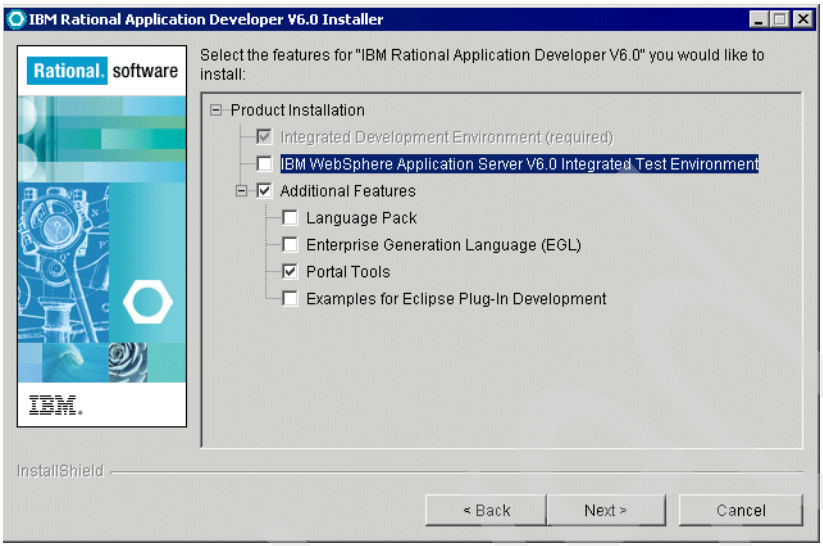


Figure 12-10 RAD - Options select

6. Figure 12-11 shows a summary of the selected options. Review the options and click **Next** to continue.

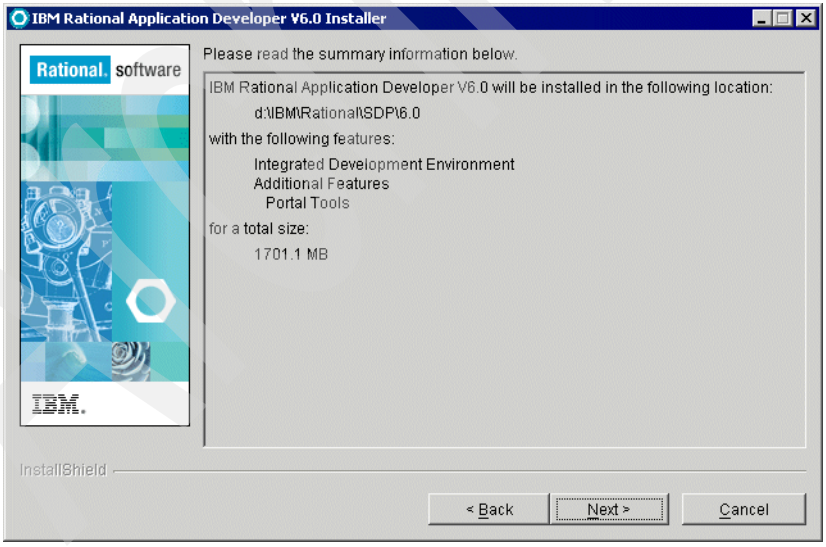


Figure 12-11 RAD summary

7. After you have installed the core product, you can install the Agent Controller (Figure 12-12). You can restart the installation launchpad at anytime if you want to install the Agent Controller later.

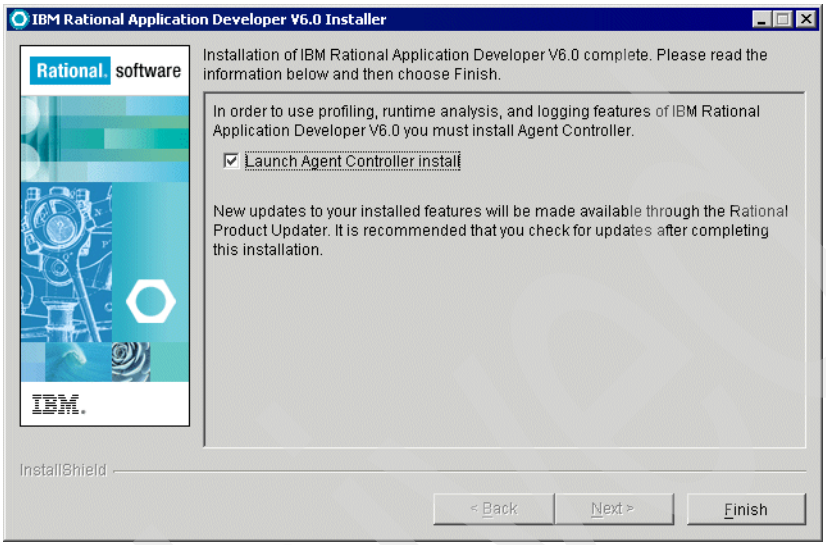


Figure 12-12 RAD, installing the Agent Controller

Tip: We recommend that you apply the available fixpacks to the RAD installation. In the examples in this book, we use V6.0.0.1 Interim Fix 002.

12.4.2 Installing the API Java files

When the installation is complete, RAD includes the necessary shared libraries for WebSphere Application Server and WebSphere Portal Server. The installation can also include optional installed test environments for these servers. If you install the WebSphere Portal 5.1 test environment, the shared libraries for IBM Workplace Web Content Management are *not* installed. You have to install these IBM Workplace Web Content Management API .jar files manually from an existing full installation (for example, your local development environment). We recommend that you copy all files to a new directory below the runtimes directory (for example, D:\IBM\Rational\SDP\6.0\runtimes\IWWCM).

You need the .jar files that contain the IBM Workplace Web Content Management server and the JSP tag library definition. The jar files are stored in the following directory:

<*Drive*>:\<*Path*>\WebSphere\PortalServer\wcm\shared\app

The tag library definition is named wcm.tld and is located under the IWWCM.war directory as follows:

<*Drive*>:\<*Path*>\WebSphere\AppServer\installedApps\<*Node*>\wcm.ear\ ilwwcm.war\WEB-INF\tld

Copy all files to the new directory below the runtimes directory. If you copy all files to this directory, the directory should contain the files as illustrated in Figure 12-13.

ilwwcm-api.jar	167 KB	Executable Jar File
ilwwcm-commons-copyright.jar	1 KB	Executable Jar File
ilwwcm-commons-properties.jar	4 KB	Executable Jar File
ilwwcm-commons-utils.jar	208 KB	Executable Jar File
ilwwcm-commons-version.jar	3 KB	Executable Jar File
ilwwcm-commons-xmlpersistency.jar	55 KB	Executable Jar File
ilwwcm-framework.jar	1,590 KB	Executable Jar File
ilwwcm-search-core.jar	69 KB	Executable Jar File
ilwwcm-server.jar	2,951 KB	Executable Jar File
ilwwcm-syndication-core.jar	217 KB	Executable Jar File
wcm.tld	5 KB	TLD File

Figure 12-13 Java files for the IBM Workplace Web Content Management API

12.4.3 Attaching to a remote server

You can install WebSphere Portal 5.1 as an integrated test environment into RAD. However, this WebSphere Portal 5.1 test environment installation does not install the necessary IBM Workplace Web Content Management functionality. Therefore, you cannot use an integrated test environment for the IBM Workplace Web Content Management API development. Instead, you can use a full-featured WebSphere Portal server that is attached as a remote server to the RAD.

The following example explains how to attach a remote Portal server. The section does not cover the necessary configuration of the WebSphere Application Server to use its remote debug possibilities. Refer to Chapter 16 of *IBM WebSphere Portal V5 - A Guide for Portlet Application Development*, SG24-6076, for a detailed description on how to set up remote debugging. To attach a remote Portal server, do the following:

1. If needed, start the RAD.
2. Switch to a perspective that includes the Server tab. (for example, the Web or J2EE perspective).
3. Right-click the background of the Server tab, and select **New Server** in the context menu (Figure 12-14).

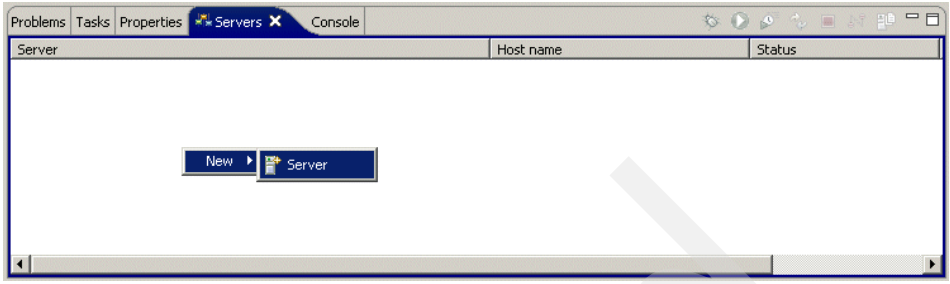


Figure 12-14 Context menu, New Server

4. A new window displays to allow you to define the new server configuration (Figure 12-15). Do the following:
   a. Enter or select the full qualified host name of your remote WebSphere Portal server.
   b. Select **WebSphere Portal 5.1 Server Attach** as the server type.
   c. Click **Next** to continue.

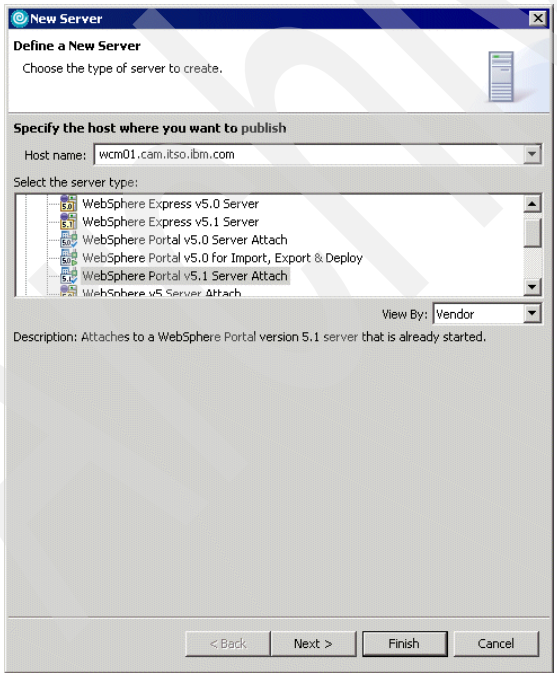


Figure 12-15 New server dialog

5. The Server Ports windows displays (Figure 12-16). if you have not changed the ports configuration of your WebSphere Portal, keep the default settings by clicking **Next**.

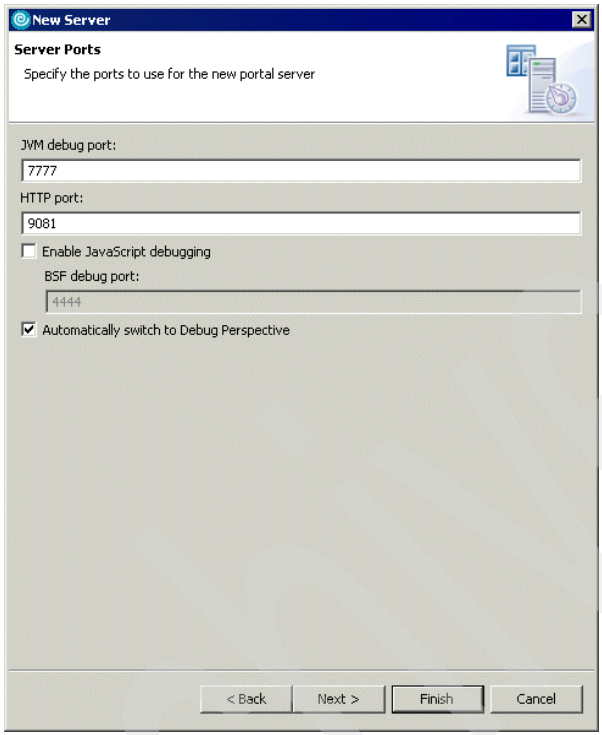


Figure 12-16 WebSphere Portal setting, server ports

6. A new dialog box displays to allow you to define your WebSphere Portal server settings (Figure 12-17 on page 500). Generally, you need to specify only the name and password of your Portal user.

Archived

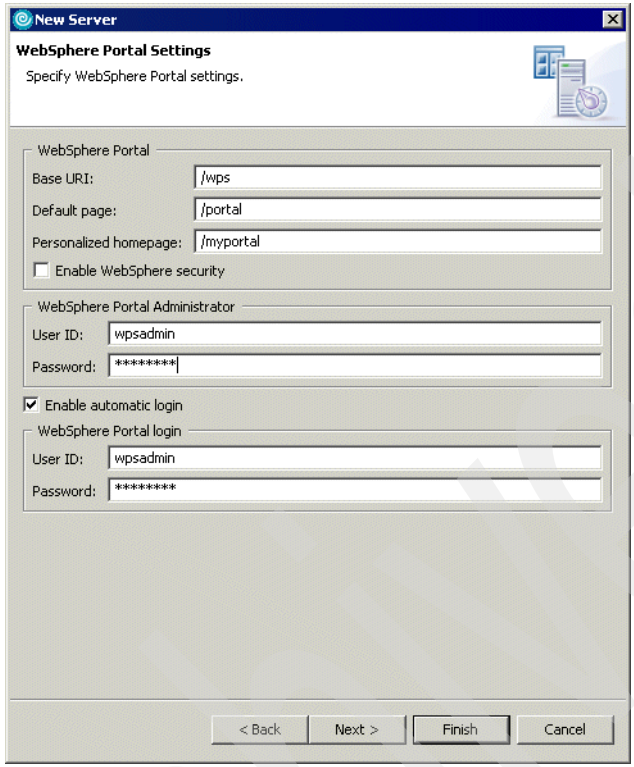


Figure 12-17 WebSphere Portal setting, preview server

7. Click **Finish**.

The attached remote server is added to your server list, as shown in Figure 12-18.

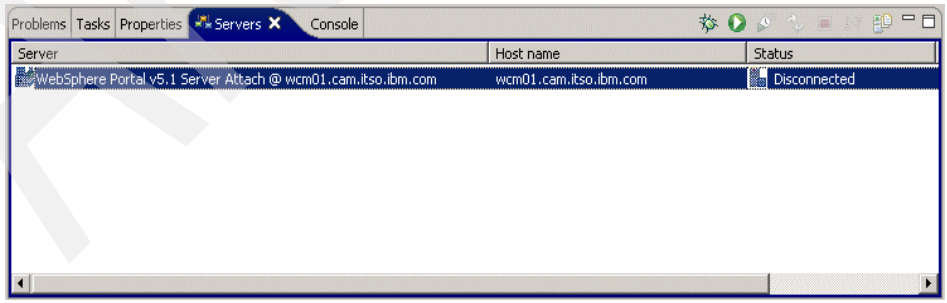


Figure 12-18 Attached server

12.5 Examples of API development

This section contains two simple examples that illustrate development with the IBM Workplace Web Content Management API.

The first example provides good starting point for developers who are not familiar with the development of IBM Workplace Web Content Management portlets.

The second example integrates JSP component into an existing IBM Workplace Web Content Management Web site or portal. This example does not use the API itself.

Note: The IBM Workplace Web Content Management installation contains further API examples. You can find the code for these examples at:

<*Drive*>:\<*Path*>\PortalServer\wcm\samples\API\doc\wcm-5-1-0-1-api-samples.htm

12.5.1 Developing a simple Portlet

This example creates a simple portlet that displays the content of a specified site area. The portlet is based on the basic portlet. It contains an edit mode and, of course, the view mode. In the edit mode, the user can select a site area. The selected site area is saved as portlet data. The view mode fetches the data from the portlet data and displays the content that is associated with the selected site area.

Archived

Figure 12-19 shows the sequence of actions in this example.

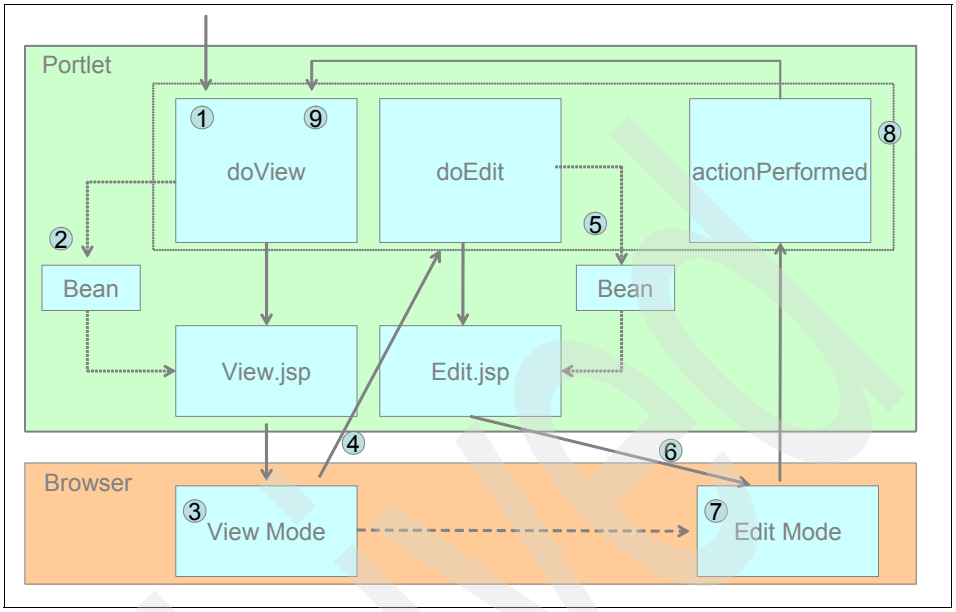


Figure 12-19 SimpleDisplay Portlet action sequence

The steps in this example are as follows:

1. The `doView` method is executed.
2. A JSP is called to render an initial screen.

   A message is obtained as JavaBean from the request object. JavaBeans™ are a special type of Java class that contain the business logic of the application. They are used to store and process data temporarily and to access backend resources such as databases.
3. The Portlet View mode screen is shown in the browser window.
4. The user clicks **Edit** to switch to Edit mode.
5. The `doEdit` method is executed. A message is obtained from the request object.
6. The Edit mode screen displays.
7. The user selects the desired site area and clicks **Submit**.
8. The `actionPerformed` method is executed to process the action. A message is stored in the PortletData.
9. The `doView` method is executed to complete the cycle.

Note: The IBM Workplace Web Content Management is integrated in WebSphere Portal since version 5.1. Therefore, we expect that some readers of this book who have already experience with IBM Workplace Web Content Management might not have extensive experience in the portlet development. For that reason, we start our IBM Workplace Web Content Management portlet development example from scratch.

Our goal is to provide a complete example that every developer can follow. If you have skills in portlet development, you can move through the first step quickly. We highly recommend that you read *IBM Rational Application Developer V6 - Portlet Application Development and Portal Tools*, SG24-6681, for a detailed introduction into portlet development.

Step 1: Creating a basic portlet

To create a basic portlet, do the following:

1. Start RAD if necessary.
2. Click **File** → **New** → **Portlet Project** to create a new portlet project.

   The New Portlet Project window appears (Figure 12-20).

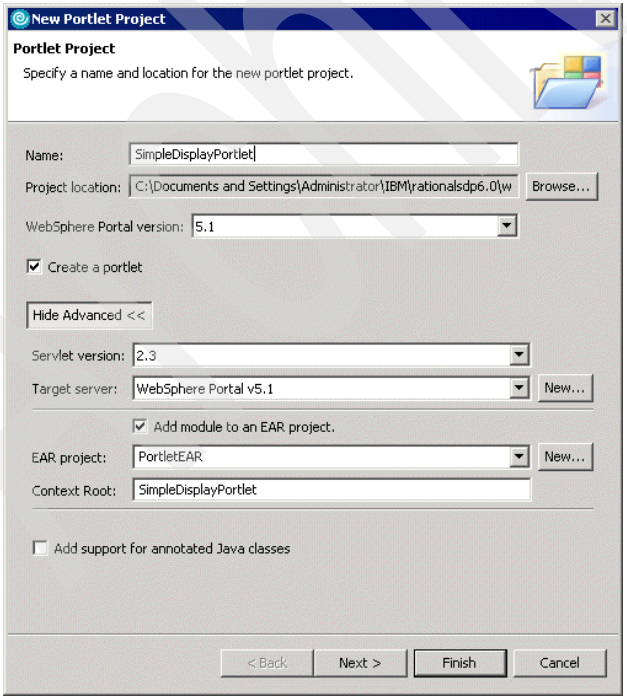


Figure 12-20 New Portlet Project window

3. Enter the following information:
   a. Name: `SimpelDisplayPortlet`
   b. WebSphere Portal version: `5.1`
   c. Servlet version: `2.3`
   d. Target server: `WebSphere Portal V5.1`
4. Select the following options:
   a. Create a portlet
   b. Add module to EAR project option
5. Select an EAR project or create a new project.
6. Click **Next**.

   The Portlet Type windows displays (Figure 12-21).

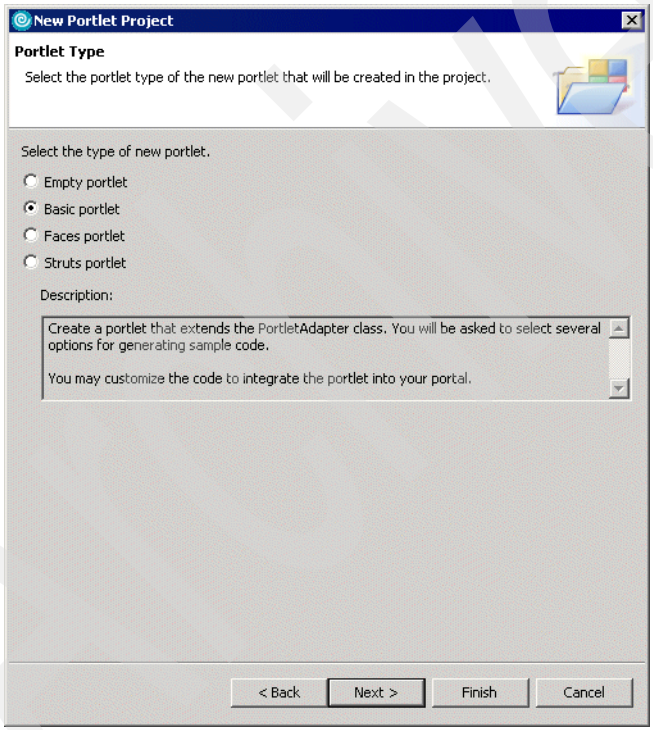


Figure 12-21 Portlet Type window

7. Select **Basic portlet** and click **Next**.

The Features window displays (Figure 12-22).

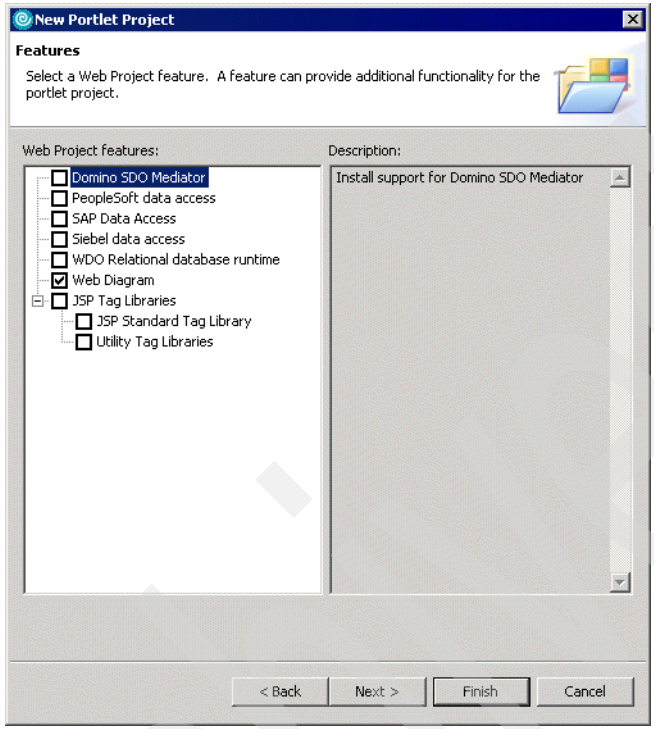


Figure 12-22 Features window

8. Keep the default settings and click **Next**.

Archived

The Portlet Settings window displays.

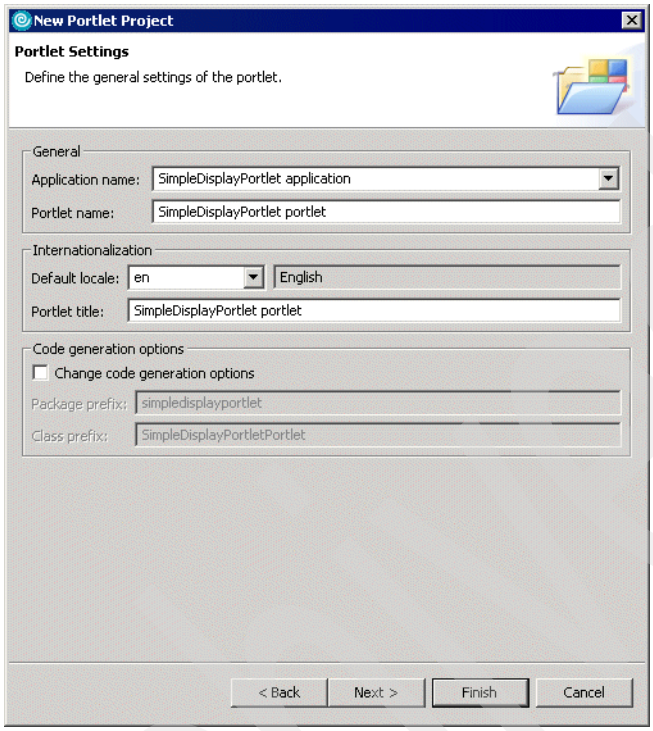


Figure 12-23 Portlet Settings window

9. Keep the default settings again and click **Next**.

Archived

The next window (Figure 12-24) contains the event handling settings.

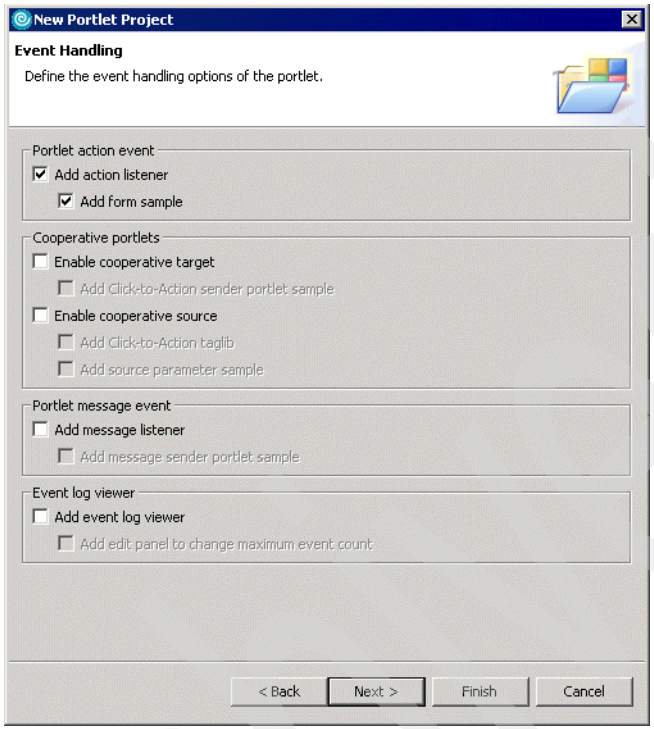


Figure 12-24 Event Handling window

10. Select **Add action listener** and **Add from sample**.
11. Click **Next**.

Archived

The Single Sign-On window appears.

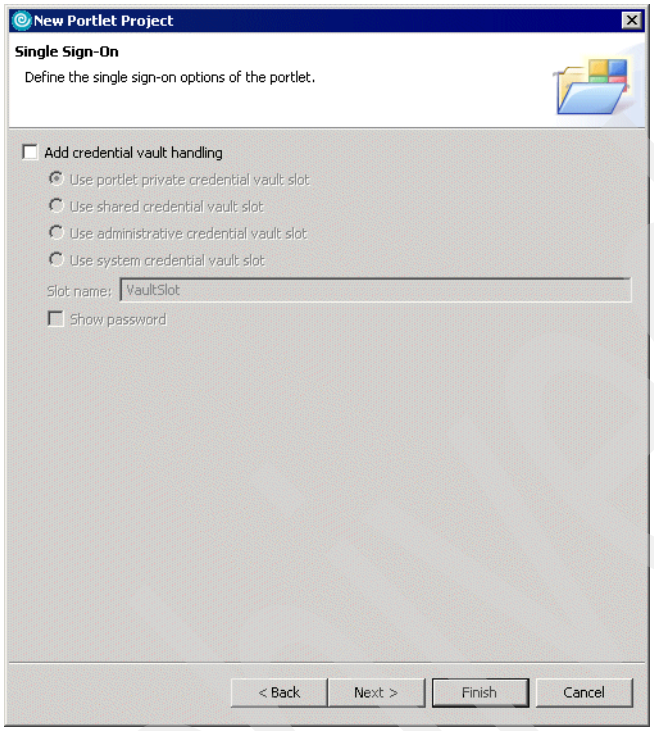


Figure 12-25 Single Sign-On window

12. Keep the default settings because you do not use any credentials in this example.
13. Click **Next**.

Archived

14. Select **Add edit mode** to add the sample code for the portlet edit mode to the new project (Figure 12-26).

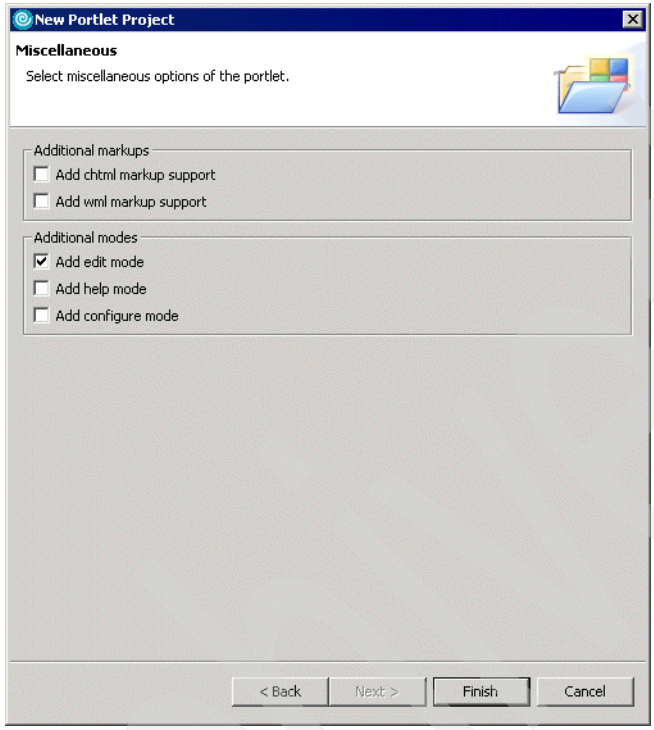


Figure 12-26 Miscellaneous windows

15. Click **Finish**.

Archived

The Portal-Tool wizard creates a new portlet project. You can find the project under Dynamic Web Projects in the Web perspective. Figure 12-27 illustrates the new project expanded in the Web perspective.

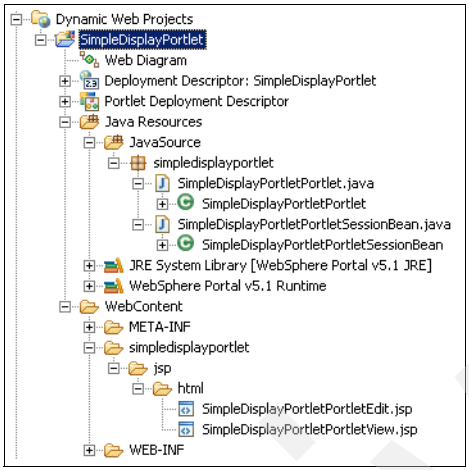


Figure 12-27 Web perspective, new portlet project.

16. Make sure that your development environment is set up correctly by selecting the new project. Right-click to open the context menu and select **Run** → **Run on server...** as illustrated in Figure 12-27.

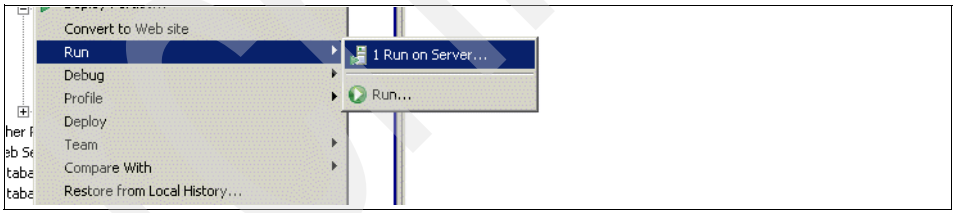


Figure 12-28 Run on Server...

Archived

The Server Selection window appears (Figure 12-29).

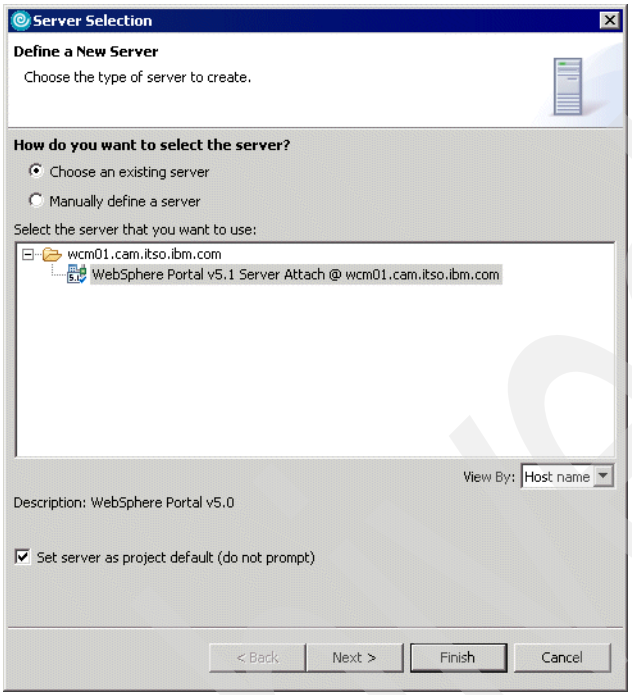


Figure 12-29 Select server

17. Select the WebSphere Portal Server V5.1 that you set up in 12.4, “Setting up the development environment” on page 491. Optionally, you can select **Set server as project default**.
18. Click **Finish**.

Archived

After the portlet deploys to the WebSphere Portal server, the Web browser shows the basic portlet within the portlet preview. If the portlet is displayed as illustrated in Figure 12-30 the environment is set up correctly.

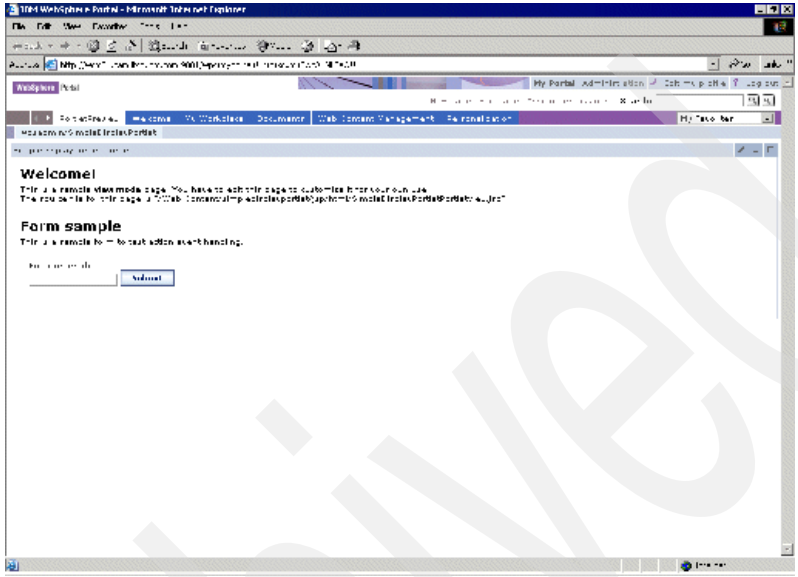


Figure 12-30 Portlet preview

Step 2: Extending the Java Build path

Before you can develop the IBM Workplace Web Content Management API portlet example, you have to extend the Java Build path and add the API libraries. To do this, follow these steps:

1. Expand the Dynamic Web Project Category of the Web perspective.
2. Right-click the SimpleDisplayPortlet project and select **Properties** from the context menu.

Archived

The Properties for SimpleDisplayPortlet appears (Figure 12-31).

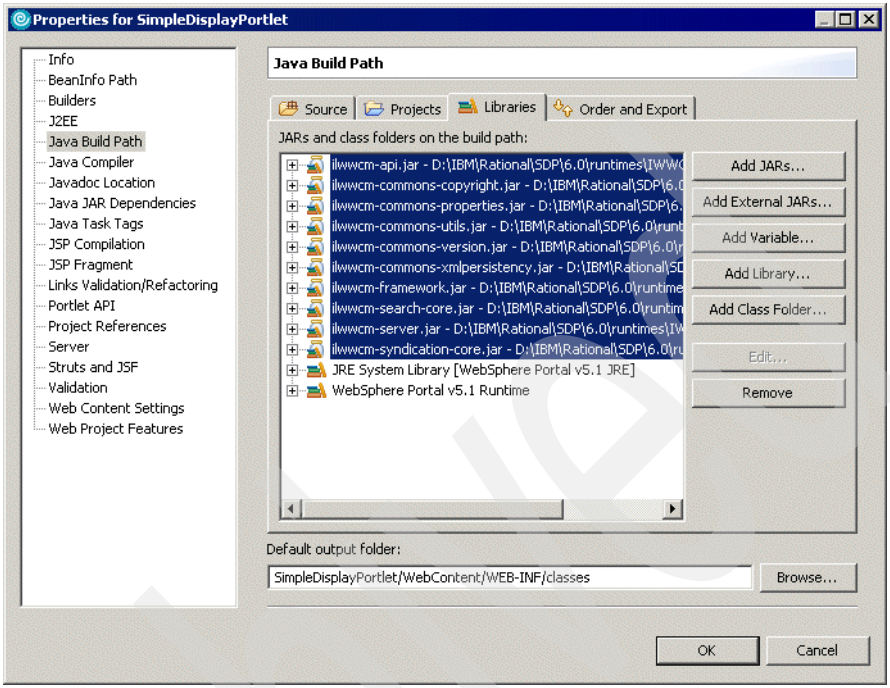


Figure 12-31 Java Build Path Settings

3. Select **Java Build Path** → **Libraries**.
4. Select **Add External JARs...**

5. Select all the IBM Workplace Web Content Management .jar files in your local runtimes directory of your RAD installation. 12.4.2, “Installing the API Java files” on page 496 provides details about these files.

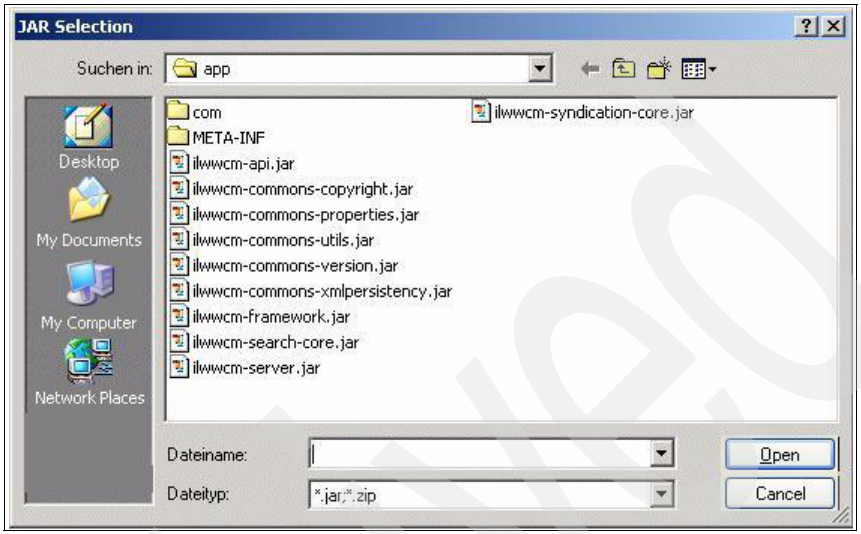


Figure 12-32 Select external JAR files

6. Click **OK**.
7. Extend the folders WebContent → WEB-INF and create a new folder named tld below the WEB-INF folder using the **File** → **New** → **Folder** menu. This new directory contains the JSP tag library definition file.

Archived

8. Select the new folder and import this definition file by selecting **File** → **Import**.
9. Select File system and click **Next**.

   The Import windows appears (Figure 12-33).

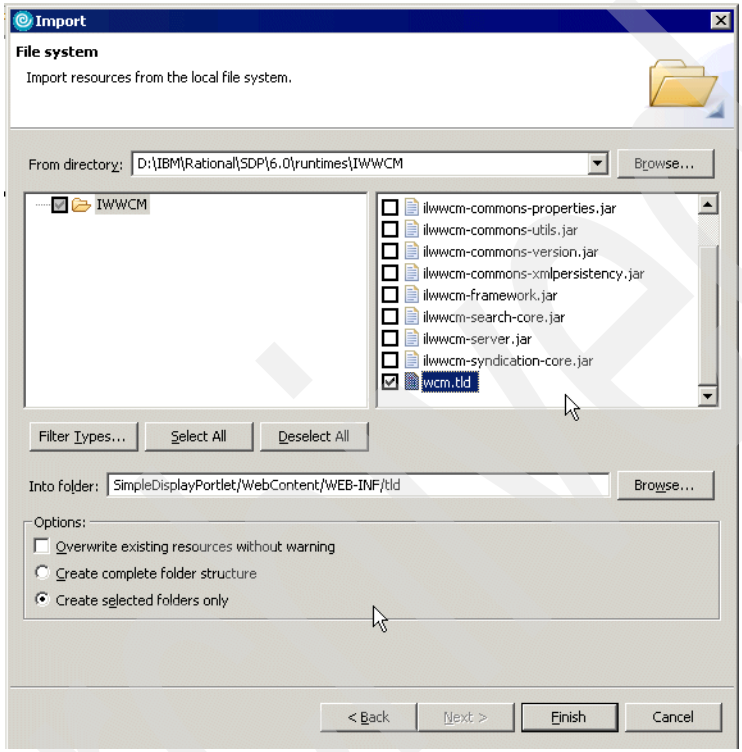


Figure 12-33 Importing wcm.tld

10. Browse to your runtime directory (as described in 12.4.2, “Installing the API Java files” on page 496) and select the wcm.tld file.
11. Click **Finish**.

Step 3: Developing the Portlet

After you have finished all the configuration steps that are necessary to create the portlet with IBM Workplace Web Content Management functionality, you can start writing the Java code. Because you used the basic portlet option to create a new portlet project, all required Java classes and JSP are already created by the portlet wizard. The class and JSP contain sample code that you need to replace with your code.

You need to develop the following code for this example:

- JSP for the edit mode of the portlet, Edit.jsp, which allows us to select a single site area
- JSP for the view mode of our portlet, View.jsp. This page uses the IBM Workplace Web Content Management JSP tag library to display the default content of the selected site area. The path to the content is retrieved as a string from the session.
- The `doEdit` method that iterates through all sites and gathers the names of all site areas. The ArrayList is added to the session and passed to the edit JSP.
- The `actionPerformed` method that saves the selected site area to the PortalData.
- The `doView` methods that search for the default content of the selected site area, that calculate the path to the content, and that add this path as string to session.
- A JavaBean that is added to the session in edit mode.
- A JavaBean that is stored in the PortletData.

Before you can display any content document, you need to select a site area and to store this area in the PortletData. Thus, you develop the Java code in the following order:

1. SessionBean
2. `doEdit` method
3. Edit mode JSP
4. PortletData Bean
5. `actionPerformed` method
6. `doView` methods
7. View mode JSP

Archived

SessionBean

The session bean SimpleDisplayPortletPortletSessionBean stores an array list of all site area document IDs and of all corresponding content paths. In addition to these two fields, the session bean contains corresponding `get` and `set` methods. Figure 12-10 illustrates the Java code of the session bean.

Example 12-10 SimpleDisplayPortletPortletSessionBean

```
package simpledisplayportlet;
import java.util.ArrayList;

public class SimpleDisplayPortletPortletSessionBean {

   private ArrayList contentPaths = null;
   private ArrayList siteAreaIds = null;

   public ArrayList getContentPaths() {
      return contentPaths;
   }

   public void setContentPaths(ArrayList contentPaths) {
      this.contentPaths = contentPaths;
   }

   public ArrayList getSiteAreaIds() {
      return siteAreaIds;
   }

   public void setSiteAreaIds(ArrayList siteAreaIds) {
      this.siteAreaIds = siteAreaIds;
   }
}
```

doEdit method

The first method that you develop is the `doEdit` method. This method gathers all document IDs and content paths of all site area's and stores both lists in the session bean. The lists are stored as array list. Thus, you need to import the array list class using the import statement that is shown in the first line of Example 12-11. The other two import statements in this example allow you to program using the IBM Workplace Web Content Management API.

Example 12-11 SimpleDisplayPortletPortlet class import

```
import java.util.ArrayList;
import com.ibm.workplace.wcm.api.*;
import com.ibm.workplace.wcm.api.exceptions.*;
```

Archived

Example 12-12 shows the entire source code of the `doEdit` method. The method does the following:

1. Before you can access any IBM Workplace Web Content Management functionality, you need to obtain a workspace for the current user.
2. Use the `Workspace.findByType` method to create a document ID iterator that contains the document IDs of all site documents within the site framework.
3. The `doEdit` method should gather all site areas of each site. Therefore, you use the `getAllChildren` method to create a second iterator. This second iterator contains the document IDs of all child documents below the current site. The children are either site area or content. You only continue for containers that match the document type SiteArea.
4. To show the hierarchal list of site areas in a user friendly way, you want to display the site area with the full path. You can build the full path by iterating through all parent site areas of the current site area. Each element in this path is concatenated with a forward slash.
5. At the end, you add the created array lists to the session bean.

Example 12-12 doEdit method

```
public void doEdit(PortletRequest request, PortletResponse response) throws PortletException,
IOException {
   SimpleDisplayPortletPortletSessionBean sessionBean = getSessionBean(request);
   //1. get the workspace
   Workspace ws = null;
   Repository repository = null;
   try {
      repository = WCM_API.getRepository();
      ws = repository.getWorkspace((java.security.Principal)request.getUser());
   } catch (ServiceNotAvailableException e) {
      e.printStackTrace();
   } catch (OperationFailedException e) {
      e.printStackTrace();
   }

   ArrayList contentPaths = new ArrayList();
   ArrayList siteAreaIds = new ArrayList();

   //2. get all sites
   DocumentIdIterator siteIterator = ws.findByType(DocumentTypes.Site);
   Site siteDocument = null;
   while (siteIterator.hasNext()){
      try {
         siteDocument = (Site)ws.getById((DocumentId) siteIterator.next());
      } catch (DocumentRetrievalException e1) {
         e1.printStackTrace();
      } catch (AuthorizationException e1) {
```

```
            e1.printStackTrace();
        }

        //3. get all site areas for the current site
        DocumentIdIterator siteareaIterator = siteDocument.getAllChildren();
        DocumentId siteareaID = null;
        while (siteareaIterator.hasNext()){
            siteareaID = (DocumentId)siteareaIterator.next();
            if (siteareaID.isOfType(DocumentTypes.SiteArea)) {
                try {
                    //4. Create aite area path
                    StringBuffer contentPath = new StringBuffer();
                    SiteArea sitearea;
                    sitearea = (SiteArea)ws.getById(siteareaID);
                    contentPath.append("/"+sitearea.getName());
                    DocumentId parentID = sitearea.getParent();

                    while (parentID!=null) {
                        if (parentID.isOfType(DocumentTypes.SiteArea)) {
                            SiteArea siteareaParent = (SiteArea)ws.getById(parentID);
                            contentPath.insert(0,"/"+siteareaParent.getName());
                            parentID = siteareaParent.getParent();
                        } else {
                            contentPath.insert(0,"/"+siteDocument.getName());
                            parentID = null;
                        }
                    }
                    siteAreaIds.add(siteareaID.toString());
                    contentPaths.add(contentPath.toString());
                } catch (DocumentRetrievalException e2) {
                    e2.printStackTrace();
                } catch (AuthorizationException e2) {
                    e2.printStackTrace();
                }
            }
        }
    }
    //5. set the session bean
    repository.endWorkspace();
    sessionBean.setContentPaths(contentPaths);
    sessionBean.setSiteAreaIds(siteAreaIds);

    getPortletConfig().getContext().include(EDIT_JSP+getJspExtension(request), request,
response);
// The Source will be with you, always.
}
```

Archived

Edit mode JSP

The edit mode JSP, Edit.jsp, retrieves the two army lists from the session bean. After that, you go through the lists and build an HTML `SELECT` field. Each site area creates one `OPTION` entry in the select box. The select box shows all the Web site paths and uses the string representation of the corresponding document ID as value, as shown in Example 12-13

Example 12-13 SimpleDisplayPortletPortletEdit.jsp

```
<%@ page session="false" contentType="text/html"
import="org.apache.jetspeed.portlet.*,simpledisplayportlet.*,java.util.*" %>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>

<% SimpleDisplayPortletPortletSessionBean sessionBean =
(SimpleDisplayPortletPortletSessionBean)portletRequest.getPortletSession().getAttribute(SimpleD
isplayPortletPortlet.SESSION_BEAN); %>
<H2>Please select a sitearea</H2>
<form method="POST" action="<portletAPI:createReturnURI><portletAPI:URIAction
name='<%=SimpleDisplayPortletPortlet.EDIT_ACTION%>'/></portletAPI:createReturnURI>">
Site Area:
<SELECT name="siteArea">
<%
   if (sessionBean != null) {
      ArrayList paths = sessionBean.getContentPaths();
      ArrayList ids = sessionBean.getSiteAreaIds();
      String value = "";
      String name = "";
      if (paths != null && ids != null) {
         Iterator values = ids.iterator();
         Iterator names = paths.iterator();
         while (names.hasNext() && values.hasNext()) {
            value = (String) values.next();
            name = (String) names.next();
%>
<OPTION value="<%=value%>"> <%=name%></OPTION>
<%
         }
      }
   }
%>
</SELECT><BR>
<INPUT type="submit" name="Submit" value="Submit">
</form>
```

Archived

PortletData Bean

The PortletData bean contains the document ID of the selected site area. The bean is used to store the selected data to the PortletData. The class that represents the Java bean needs to implement the Serializable interface. Instead of modifying existing code, you need to create a new class for this Java bean. As shown in Example 12-14, the class contains only a private string field and a `get` and `set` method.

Example 12-14 ContentBean class

```
package simpledisplayportlet;

import java.io.Serializable;

public class ContentBean implements Serializable {
   private String siteAreaId = null;

   public String getSiteAreaId() {
      return siteAreaId;
   }

   public void setSiteAreaId(String siteAreaId) {
      this.siteAreaId = siteAreaId;
   }
}
```

actionPerformed method

The final development step in the edit process of the portlet is the `actionPerformed` method. So far, you have developed the code that allows you to create a list of site areas and to pass this list to the edit JSP. You have created the JSP that displays in edit mode and the bean that stores the data. The missing step is the function that reads the data and stores it in the created bean. Example 12-15 shows the `actionPerformed` method.

The actionPerformed method is executed when the edit JSP is saved. You retrieve the value of the siteArea field from the request and store the value in the PortletData bean.

Example 12-15 actionPerformed method

```
public void actionPerformed(ActionEvent event) throws PortletException {
   if( getPortletLog().isDebugEnabled() )
      getPortletLog().debug("ActionListener - actionPerformed called");

   String actionString = event.getActionString();
   PortletRequest request = event.getRequest();
```

Archived

```
   SimpleDisplayPortletPortletSessionBean sessionBean = getSessionBean(request);

   if( EDIT_ACTION.equals(actionString) ) {
      if( request.getParameter("siteArea")!=null ) {
         PortletData data = request.getData();
         if( data!=null ) {
            ContentBean cb = new ContentBean();
            cb.setSiteAreaId(request.getParameter("siteArea"));
            data.setAttribute(NewPortletPortlet.WCM_CONTENT, cb);
            try {
                 data.store();// save to data store
            }
            catch (IOException ioe) {
               if( getPortletLog().isErrorEnabled() )
                  getPortletLog().error("Error on PortletData.store(): "+ioe.getMessage());
            }
         }
      }
   }
}
```

The PortletData bean can be accessed by using a defined key. The following example shows the definition of this key as a static string:

```
public static final String WCM_CONTENT = "contentPath";
```

The doView method

You now need to implement the part that shows the content in the portlet, as shown in the code example in Example 12-16 on page 523. The important steps within this function are the following:

1. You need to access the workplace for the current user again.
2. You retrieve the content bean from the PortletData. If the content bean cannot be retrieved, an error message is generated, and the `doView` method is left.
3. The SiteArea document is searched by the document ID that is stored in the content bean.
4. The view JSP can access content documents only by their path. Therefore, you go through all parent site area documents of this content document and create the path.
5. You add the generated path as session bean to the request.
6. You invoke the JSP to render.

Archived

Example 12-16 The doView method

```
public void doView(PortletRequest request, PortletResponse response) throws PortletException,
IOException {
   SimpleDisplayPortletPortletSessionBean  sessionBean = getSessionBean(request);
   if( sessionBean==null ) {
      response.getWriter().println("<b>NO PORTLET SESSION YET</b>");
      return;
   }
   //1. get the workspace
   Workspace ws = null;
   Repository repository = null;
   try {
      repository = WCM_API.getRepository();
      ws = repository.getWorkspace((java.security.Principal)request.getUser());
   } catch (ServiceNotAvailableException e) {
      e.printStackTrace();
   } catch (OperationFailedException e) {
      e.printStackTrace();
   }
   //2. get the site area from the content bean
   ContentBean cb =
(ContentBean)request.getData().getAttribute(SimpleDisplayPortletPortlet.WCM_CONTENT);
   if( cb==null ) {
       response.getWriter().println("<b>NO CONTENTBEAN YET</b>");
      return;
   }
   try {
      //get the defaultcontent for the site area
      SiteArea sitearea = (SiteArea)ws.getById(ws.createDocumentId(cb.getSiteAreaId()));
      Site site = (Site)ws.getById(sitearea.getParent());
      Content content = (Content)ws.getById(sitearea.getDefaultContent());

      //create the path
      StringBuffer contentPath = new StringBuffer();
      contentPath.append("/"+sitearea.getName());
      contentPath.append("/"+content.getName());

      DocumentId parentID = sitearea.getParent();

      while (parentID!=null) {
         if (parentID.isOfType(DocumentTypes.SiteArea)) {
            SiteArea siteareaParent = (SiteArea)ws.getById(parentID);
            contentPath.insert(0,"/"+siteareaParent.getName());
            parentID = siteareaParent.getParent();
         } else {
            contentPath.insert(0,"/"+site.getName());
            parentID = null;
         }
```

Archived

```
      }

      //5. add the pathstring to the request
      request.setAttribute(SimpleDisplayPortletPortlet.WCM_CONTENT, contentPath.toString());
   } catch (DocumentRetrievalException e1) {
      e1.printStackTrace();
   } catch (AuthorizationException e1) {
      e1.printStackTrace();
   } catch (DocumentIdCreationException e1) {
      e1.printStackTrace();
   } catch (PropertyRetrievalException e) {
      e.printStackTrace();
   }
       //6. Invoke the JSP to render
      getPortletConfig().getContext().include(VIEW_JSP+getJspExtension(request), request,
response);
}
```

View mode JSP

The View mode JSP retrieves the SessionBean from the request (Example 12-17). The IBM Workplace Web Content Management workspace is set for the current user, and the context is set to the path that was stored in the SessionBean. Finally, the content is rendered.

Example 12-17 SimpleDisplayPortletPortletView.jsp

```
<%@ page session="true" contentType="text/html" import="java.util.*, simpledisplayportlet.*,
java.security.Principal"%>

<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ taglib uri="/WEB-INF/tld/wcm.tld" prefix="wcm" %>
<portletAPI:init/>

<% SimpleDisplayPortletPortletSessionBean sessionBean =
(SimpleDisplayPortletPortletSessionBean)portletRequest.getPortletSession().getAttribute(SimpleD
isplayPortletPortlet.SESSION_BEAN); %>

<% String content =
(String)portletRequest.getAttribute(SimpleDisplayPortletPortlet.WCM_CONTENT); %>
<% String appPath =
"http://"+portletRequest.getServerName()+":"+portletRequest.getServerPort()+"/wps/wcm"; %>

<wcm:initworkspace user="<%= (java.security.Principal)portletRequest.getUser() %>" />
<wcm:setExplicitContext wcmServletPath="/connect" wcmWebAppPath="<%=appPath%>"
path="<%=content%>">
<%= error %>
</wcm:setExplicitContext>
<wcm:content pageDesign="PortalPageLayout" >Content error - <%= error %></wcm:content>
```

Note: We assume that you use this example with the River Bend example Web site and use the PortalPageLayout as the page design during the content rendering. If you use this example in a different context, you use a different page design. You also can leave this JSP parameter empty and use the default presentation template of the content.

Step 4: Previewing the Portlet

Finally, you can preview the portlet by following these steps:

1. Right-click the SimpleDisplayPortlet project and chose **Run** → **Run on Server...** from the context menu.
2. Select the previously configured WebSphere Portal Server (as described in 12.4.3, “Attaching to a remote server” on page 497).

   The new portlet is deployed to WebSphere Portal. During the deployment, a new portal preview page is created, and the portlet is added to this page.

   After these steps the browser shows and displays the new portlet. Because you have not selected a site area yet, the `NO CONTENTBEAN YET` error displays.
3. Click the small wrench symbol to switch to edit mode.
4. As shown in Figure 12-34, the Edit.jsp with the **option select box appears. Select a site area from the select box and click Submit**.

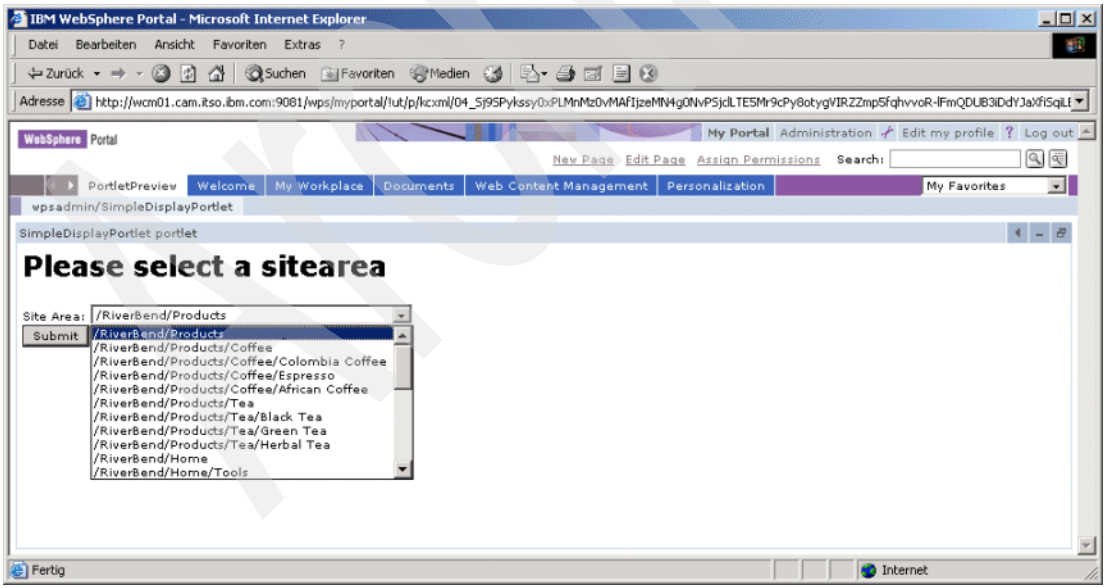


Figure 12-34 SimpleDisplayPortlet, edit mode

After you submit the selection in the Edit.jsp, the `actionPerformed` method executes, and the content bean is saved. Then, the `doView` method executes. Finally, the content is displayed in the portlet, as shown in Figure 12-35.

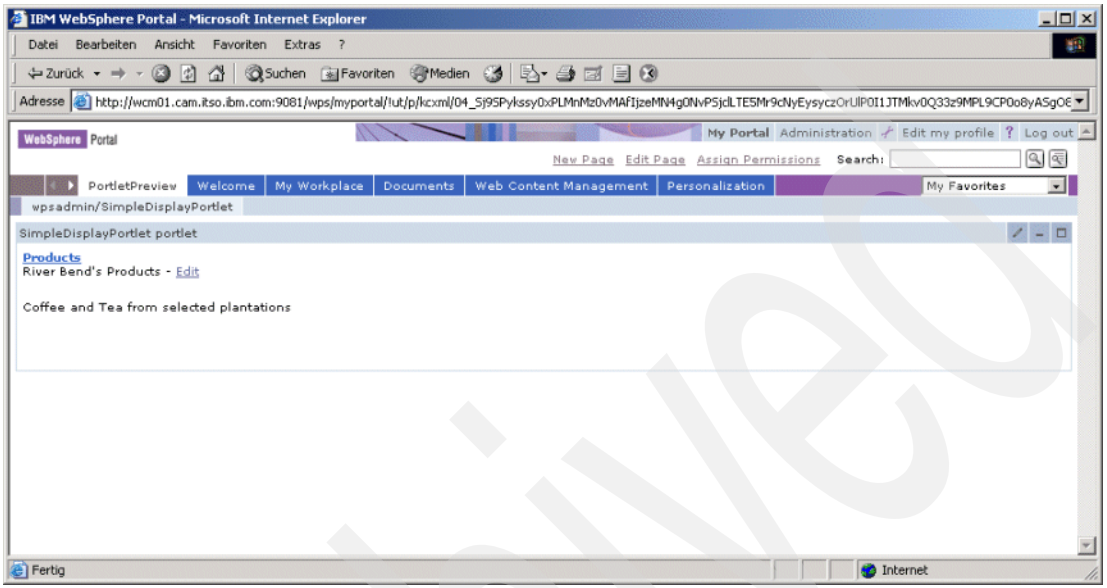


Figure 12-35 SimpleDisplayPortlet, view mode

12.5.2 Working with JSP components

This example shows the deployment and use of the JSP component feature. (You can find general information about the JSP component in 12.3.1, “JSP components” on page 486.) Within this example, we extend the content presentation of the River Bend example to demonstrate how you can use JSP components to add dynamic functionality to Web content. In addition, we discuss the necessary steps to deploy JSP components.

To keep the example as simple as possible, we add a basic time and date function to our River Bend Web site. In addition to this functionality, we create a JSP that use the java.util.Date object to print the current data and time. Figure 12-18 shows the code for this JSP.

Example 12-18 JSP component

```
<p>The current date is:
<%=new java.util.Date().toString()%></p>
```

To begin, create a simple text file named time.jsp and store this file in the JSP directory below the Web content management .war file:

<*Drive*>:\<*Path*>\WebSphere\AppServer\installedApps\<*Node*>\wcm.ear\ilwwcm.war\jsp

Then, integrate the new JSP into the content presentation as follows:

1. Log in to IBM Workplace Web Content Management and open the Web Content Authoring portlet if necessary.
2. Expand Design and Development.
3. Click **New** and select **JSP Component**.
4. Click **OK**.
5. Enter the information about the JSP component as shown in Figure 12-36.

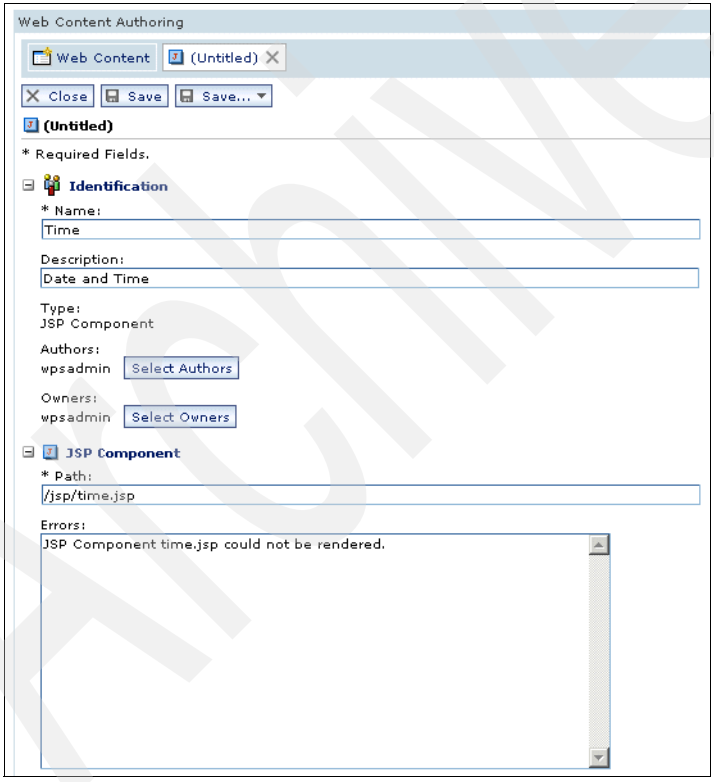


Figure 12-36 JSP component

Archived

Enter the following values:

a. Name: `Time`
b. Description: `Date and Time`
c. Path: `/jsp/time.jsp`
d. Errors: `JSP component time.jsp could not be rendered.`

The path *must* begin with a forward slash. The path represents the directory below the iwwcm.war directory. This field is used to enter an error message to display if the JSP is unavailable.

Note: The name of the JSP file is case sensitive.

Important: The JSP referenced within a JSP component must not include a reference, directly or indirectly, to the same JSP component. This includes references within IBM Workplace Web Content Management tags or the API. If it does, a loop is created and your Server will crash.

6. **Click Save** and **Close**.
7. A new JSP component category appears in Design and Development → component Library.
8. Open the Presentation Templates view and open the Two Column Table presentation.
9. Click **Edit**.

Archived

10. Add a reference to the JSP component in the presentation template HTML code. In this, example we add the `<AptrixLibCmpnt name="Time"/>` tag below the HTML table, as shown in Figure 12-37.

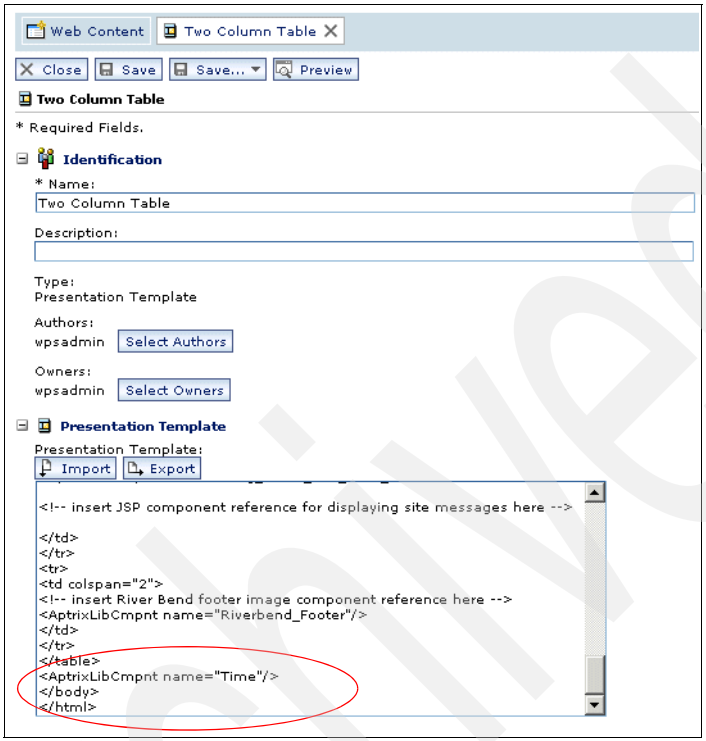


Figure 12-37 JSP Component example - Presentation Template

11. Save and Close the document.

12. Repeat steps 8 through 11 for the other both presentation templates.

Archived

13. Preview the Web site. Figure 12-38 illustrates the integrated JSP component within the River Bend Web site.

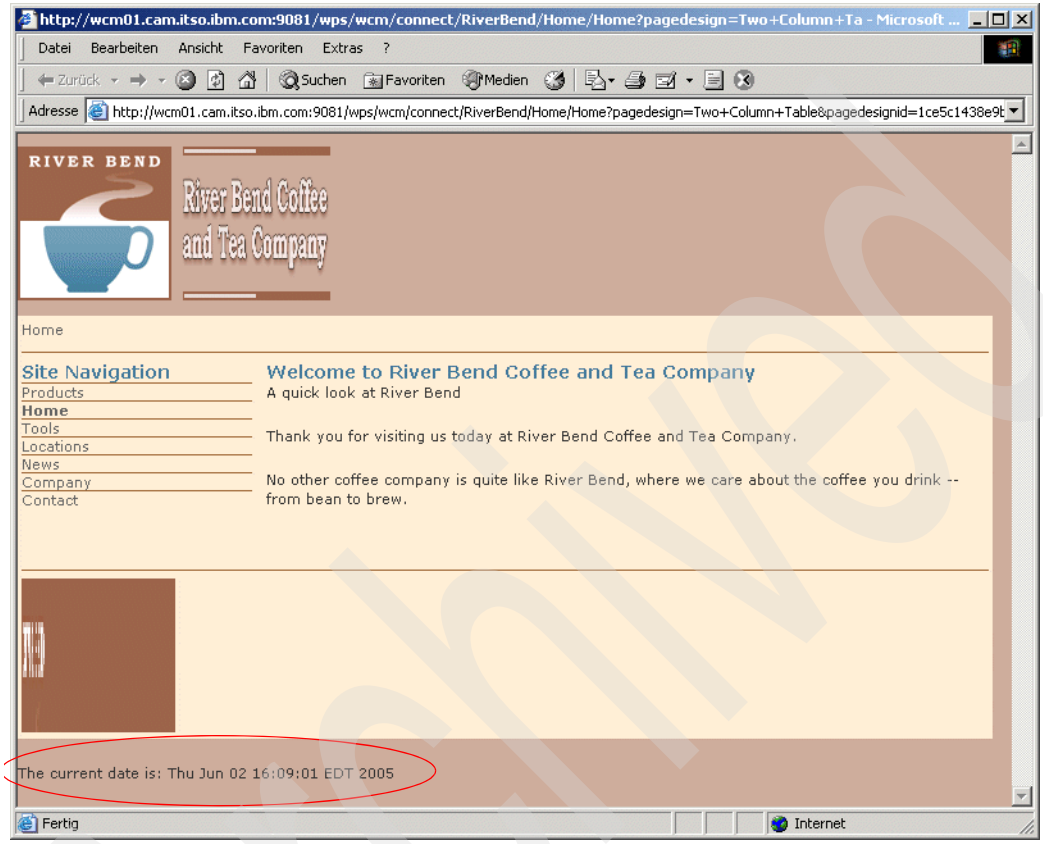


Figure 12-38 JSP Component, preview with a Web browser

12.6 Deployment considerations

Before you deploy your changes, consider the following:

- Installation

  Until version 2.0, the IBM Workplace Web Content Management API was not part of the core product installation. It was necessary to create a shared library and to add the library to the class loader of the application server. Since version 5.1, the API is installed as part of the IBM Workplace Web Content Management configuration utility. You do not need to install an API library to use the IBM Workplace Web Content Management capability.

Tip: Refer to the developerWorks article “Setting up a debug environment for IBM Workplace Web Content Management” if you are planing to use the API in a previous version. You can find this article at:

http://www.ibm.com/developerworks/lotus/library/debug-wwcm/

- API authentication

  A user name and password form WebSphere Member Manager is required to access secured IBM Workplace Web Content Management resources. The workspace cannot be retrieved using an LTPA token.

- Deployment

  Java resources created for API access, such as JSPs, class files, jars, and so forth, are not syndicated between nodes of the WebSphere Application Server. Therefore, you need to deploy all used resources to each IBM Workplace Web Content Management node within the environment.

  The API does not support native remote access to the IBM Workplace Web Content Management objects. The Java code that uses the API must reside on the same server as IBM Workplace Web Content Management. Each node requires a full instance of IBM Workplace Web Content Management for access.

  If you want to access the IBM Workplace Web Content Management objects remotely, we recommend an approach that explores the necessary IBM Workplace Web Content Management functions as a Web service.

Archived

Archived

13

Advanced topics and examples with the API

This chapter provides advanced examples how to use the IBM Workplace Web Content Management API to create custom portlets. The chapter contains the following sections:

- Multi-lingual Content Display Portlet
- Using the Customizable Template Portlet
- Extensions to the Customizable Template Portlet

Archived

© Copyright IBM Corp. 2006. All rights reserved.

13.1 Multi-lingual Content Display Portlet

In this section, we extend our Simple Display Portlet with the multi-language capabilities that are described in Chapter 9, “Advanced integration with WebSphere Portal” on page 403. (More specifically, refer to 9.5, “Implementing a custom display Portlet” on page 418.) So far, our portlet displays the default content of the selected site area. The default content is examined in the `doView` method. To implement the discussed multi-language approach, you have to change this point to do the following:

- Get the user’s default language code through the request from the browser settings.
- Search for a content which ends with the language code within the desired site area.

To implement these changes, follow these steps:

1. Add a method to the portlet that retrieves the language code from the request. The result is returned as string. Example 13-1 shows the code of this method.

Example 13-1 getLanguageFromHeader

```
private String getLanguageFromHeader(PortletRequest request) {
   String headerLang = request.getHeader("Accept-Language");
   int dash = headerLang.indexOf("-");
   if (dash > -1) {
      return headerLang.substring(0, dash);
   }
   return headerLang;
}
```

2. Change the part of the `doView` method that searches for the default content. Example 13-2 shows a source code fragment of the current `doView` method that retrieves the default content.

Example 13-2 Initial doView code fragment

```
...
if( cb==null ) {
   response.getWriter().println("<b>NO CONTENTBEAN YET</b>");
   return;
}
try {
   SiteArea sitearea =
(SiteArea)ws.getById(ws.createDocumentId(cb.getSiteAreaId()));
   Site site = (Site)ws.getById(sitearea.getParent());
   //search the default content
   Content content = (Content)ws.getById(sitearea.getDefaultContent());
```

```
   StringBuffer contentPath = new StringBuffer();
   contentPath.append("/"+sitearea.getName());
      contentPath.append("/"+content.getName());
...
```

3. Add code that retrieves the content in the user's default language behind the single line that searches for the default content. Then, the content is replaced by the content in the user's default language, if it exists. Thus, the portlet displays the default content, if content in the user's default language is not found.

   Example 13-3 illustrates the necessary code:

Example 13-3 Final doView code fragment.

```
...
Site site = (Site)ws.getById(sitearea.getParent());

Content content = (Content)ws.getById(sitearea.getDefaultContent());
// begin multi-lingual extension
String userLang = getLanguageFromHeader(request);

DocumentIdIterator siteAreaChildren = sitearea.getChildren();
DocumentId lastSiteAreaChildId = null;
while (siteAreaChildren.hasNext()) {
   lastSiteAreaChildId = (DocumentId)siteAreaChildren.next();
   if (lastSiteAreaChildId.isOfType(DocumentTypes.Content)) {
      if (lastSiteAreaChildId.getName().endsWith("_" + userLang)) {
         content = (Content)ws.getById(lastSiteAreaChildId);
         break;
      }
   }
}
// end multi-lingual extension
StringBuffer contentPath = new StringBuffer();
...
```

13.2 Using the Customizable Template Portlet

The standard IBM Workplace Web Content Management authoring interface for creating content is quite robust and makes all options available to content authors. For some content, however, you might not use or want all the features that are available. You might instead want to streamline the authoring process so that only elements and actions that are needed to create, preview, and publish your Web content are displayed.

This section provides the step-by-step detailed information about how to deploy and configure the Customizable Template Portlet — named '*Content Custom UI* — for IBM Workplace Web Content Management 2.5 and 5.1.0.1. This Portlet, in conjunction with the sample workflow portlet named *Workplace Web Content Management Sample Workflow*, creates an easy three-click environment for Web content creation.

This section extends the River Bend Web site that you created in Chapter 7, “Building the River Bend Web site with IBM Workplace Web Content Management” on page 253. We assume that you also might want to build this example within any other context. Thus, you create a new authoring template and menu component. You can integrate this example into the River Bend Web site or any other Web site that is created with IBM Workplace Web Content Management.

The scenario

The River Bend Web site contains a Bulletin Board that can be accessed through the public Web site and through the intranet portal. The portal can be accessed by Bulletin Board authors and employees with read access to the board, as shown in Figure 13-1 on page 537.

Archived

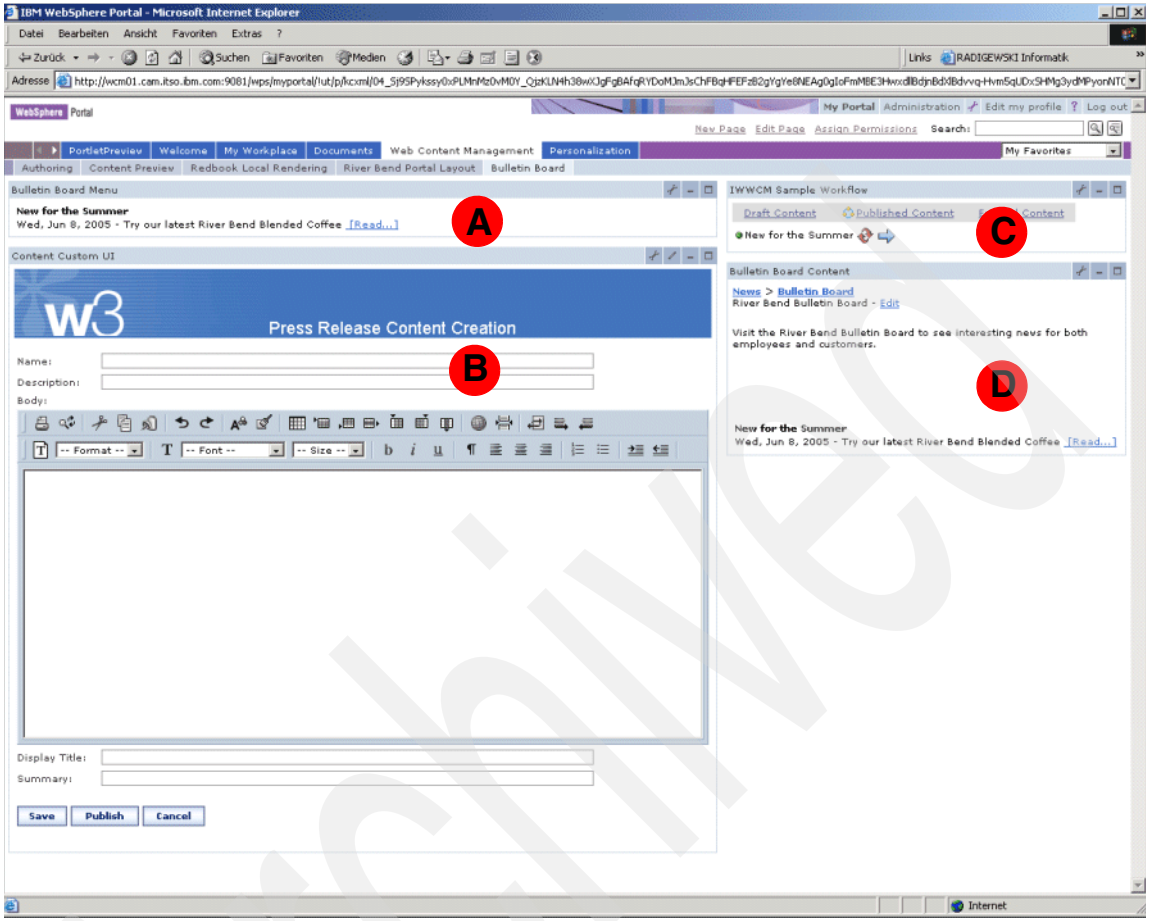


Figure 13-1 Streamlined authoring portal page

Non-authors navigate to the Bulletin Board page and only see the published Bulletin Board (A) and the selected Bulletin Board Content (D) portlets.

When authors log in to WebSphere Portal and navigates to the same Bulletin Board page, they find, in addition to A and B, the Content Custom UI (B) and Workplace Web Content Management Sample Workflow (C) portlets. Authors can add a new Bulletin Board item using these portlets in the following steps:

1. Fill Content Custom UI (B) with article information and submit the information.
2. Preview the item as it would appear in the Bulletin Board Content portlet (D).
3. Publish the item using the check mark (approve) icon in the IBM Workplace Web Content Management Sample Workflow portlet (C).

Archived

The new content is available immediately to everyone in the company.

You first create the following components:

- Authoring Template:
  - Name: `Bulletin Board`
  - Components
    - Display Title (Text Component)
    - Summary (Text Component)
    - Body (Rich Text Component)
- Presentation Template:
  - Name: `Bulletin Board`
  - Template

```
<h3><AptrixCmpnt context="current" key="Display Title"
type="content"/></h3>
<AptrixCmpnt context="current" key="Body" type="content"/>
```

Then, open the site area /RiverBend/News/Bulletin Board and create a template mapping for the new templates, as shown in Figure 13-2.

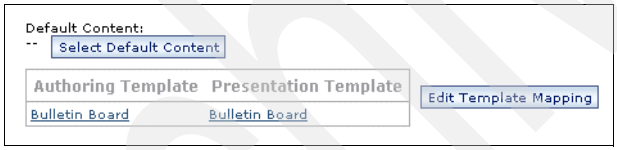


Figure 13-2 Template mapping

Note: The portlet is written for IBM Workplace Web Content Management 2.5 and WebSphere Portal 5.1.0.1. The Content Custom UI and Workplace Web Content Management Sample Workflow portlets have been tested with both of these platforms. These portlets are code samples using the IBM Workplace Web Content Management API. Thus, support is handled through normal IBM Workplace Web Content Management support of the API.

13.2.1 Installing the portlets

In preparation to install the portlets, download the following .war files to a location on your local system:

- iwwcm-custom-template-portlet_25.war (for Workplace Web Content management 2.5 or WebSphere Portal 5.1)
- iwwcm_sample_workflow_list.war

Then, follow these steps to install the portlets:

1. Navigate to the WebSphere Portal Administration → Portlet Management → Web Modules page and select **Install**.
2. Browse and select the first WAR file:

   ilwwcm-custom-template-portlet_25.war

3. Click **Next**. You see the confirmation message that is shown in Figure 13-3.

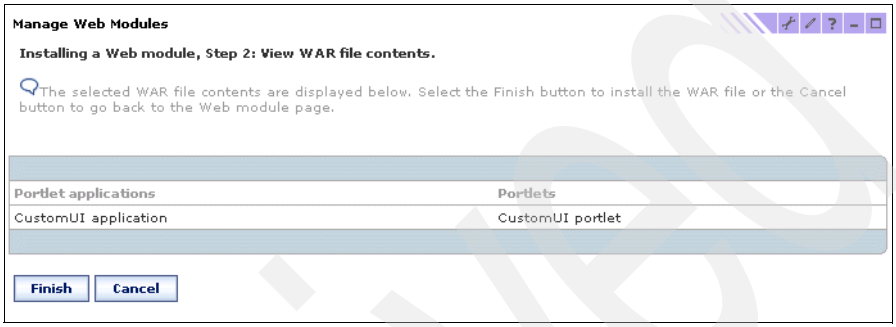


Figure 13-3 Portal Administration Install Web Modules

4. Click **Finish**.
5. Repeat these steps for the second WAR file:

   iwwcm_sample_workflow_list.war

6. Click **Next**.

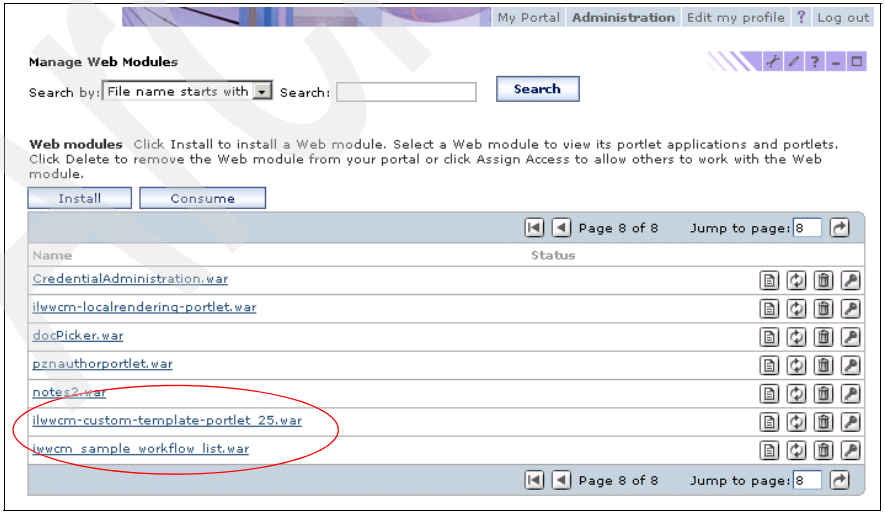


Figure 13-4 Portal Administration Install Web Modules

Archived

13.2.2 Creating a Bulletin Board page with portlets

The new portlets are added to a new Bulletin Board page, and then are configured to manage the Bulletin Board template content that is displayed on the page. To create a Bulletin Board page with portlets, follow these steps:

1. Navigate to the WebSphere Portal Administration → Portlet Management → Portlet.
2. Search for the Web Content Viewer Portlet and create a copy called Bulletin Board Menu.
3. Create a second copy and name it *Bulletin Board Content*.

> **Note:** For more information about the Web Content Viewer portlets, refer to the IBM Workplace Web Content Management Information Center at:
>
> `http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/`

4. Navigate to the WebSphere Portal Administration → Portal User Interface → Manage Pages.
5. Create an new Portlet Page called *Bulletin Board*. You can locate the page at any desired place within your Portal. If you are adding the page to the River Bend Web site. we suggest that you create the new Page below *Web Content Management*.

The new page has four portlets (two on the left and two on the right), as follows:

- Left Portlets:
  - Copy of the Web Content Viewer portlet that is configured to show a list of published Bulletin Board articles.
  - The Content Custom UI that is used for creating and editing Bulletin Board content items.
- Right Portlets
  - The Workplace Web Content Management Sample Workflow portlet that is configured to show the DeptNews template content.
  - A copy of the Web Content Viewer portlet that is configured to show document details.

Archived

6. To add the portlets to the page, select **Page Layout**.

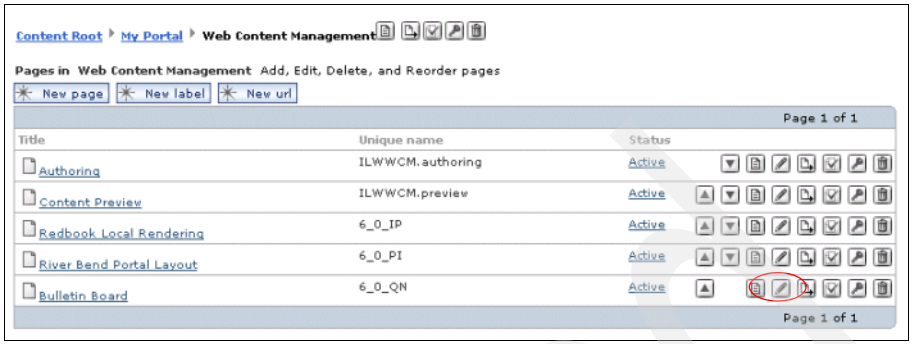


Figure 13-5 Edit page layout

7. Add the Portlets to the page. Arrange the portlets as shown in Figure 13-6 using the arrow buttons. Select **Done**.

Figure 13-6 Arrange Portlets on the Bulletin Board page

The page now has all the necessary portlets to edit quickly new IBM Workplace Web Content Management content.

Note: You can control the access to these portlets using the WebSphere Portal → Administration → Access → Resource Permissions → Portlets menu options. You need to select each of the new portlets, assign access to the user, and add the author group to that user. This action allows you to control the access to this author group only. Feel free to create an author group and assign access to these portlets for this group only.

13.2.3 Configure the Portlets

Next, you need to configure the portlets on the page. Depending on the portlet, you have to use either the edit or the configure mode.

Bulletin Board menu configuration

To use the edit mode to configure the Portlets, follow these steps:

1. Click **Configuration** on the Bulletin Board menu portlet.

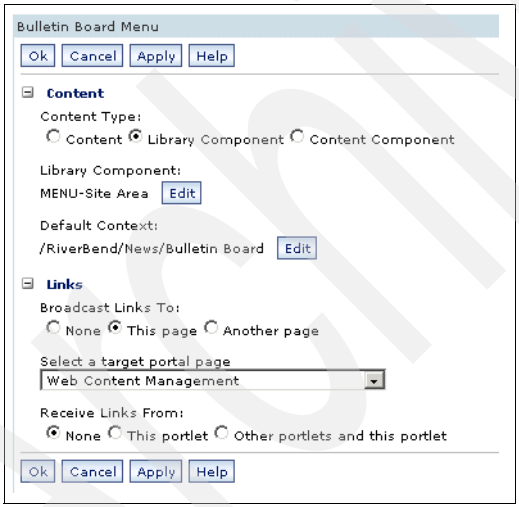


Figure 13-7 Configure the Bulletin Board menu Portlet

2. Under Content Type, select **Library Component**.
3. Chose /RiverBend/News/Bulletin Board as default content.
4. Select to broadcast links to this page.
5. Under Receive Links From, select **None**.
6. Click **OK**.

Archived

Bulletin Board content configuration

To use the configuration mode, follow these steps:

1. Click **Configuration** on the Bulletin Board menu portlet.

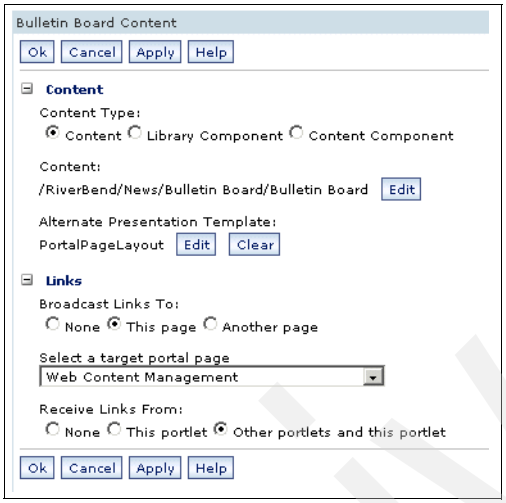


Figure 13-8 Configure the Bulletin Board content Portlet

2. Select **Content** as the Content Type.
3. Chose /RiverBend/News/Bulletin Board as default content.
4. Select Broadcast Links to **This Page**.
5. Select Receive Links From: Other Portlets and this portlet.
6. Click **OK**.

Content Custom UI Portlet configuration

For the Content Custom UI Portlet, both configure and edit modes are available. The settings in the configure mode are set for the Portlet in general and are carried with the Portlet as an instance is placed on a portal page. The configure settings are generally set by a systems administrator. The edit settings are unique to each instance of the Portlet on a page and are normally set by the content administrator. Authorization to configure and edit can be set for groups using Portal Access Control. Where settings are the same in both modes, edit overrides the configure settings.

Archived

To set the configuration setting, follow these steps:

1. On the Bulletin Board page, select the configure icon of the Content Custom UI portlet.
2. Select the Bulletin Board template, the site area /RiverBend/News/Bulletin Board/, the Development Workflow, and leave the default JSP custom template file, as shown in Figure 13-9.
3. Select **Save**.

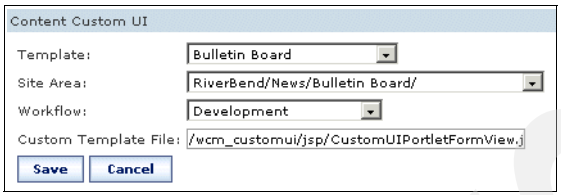


Figure 13-9 Configuring the Custom Content UI Portlet

The default JSP file CustomUIPortletFormView.jsp uses the authoring template to construct an entry field for each template attribute. A text field is used for text fields, a Rich Text editor is used for Rich Text fields as well as HTML fields, and text with browse button is used for image fields. If you need to change the entry field or placement, you can supply a custom JSP file to adjust as needed. For more information about creating a custom JSP, see “Customizing a JSP format” on page 557.

Archived

Figure 13-10 shows the Custom Content UI after configuration.

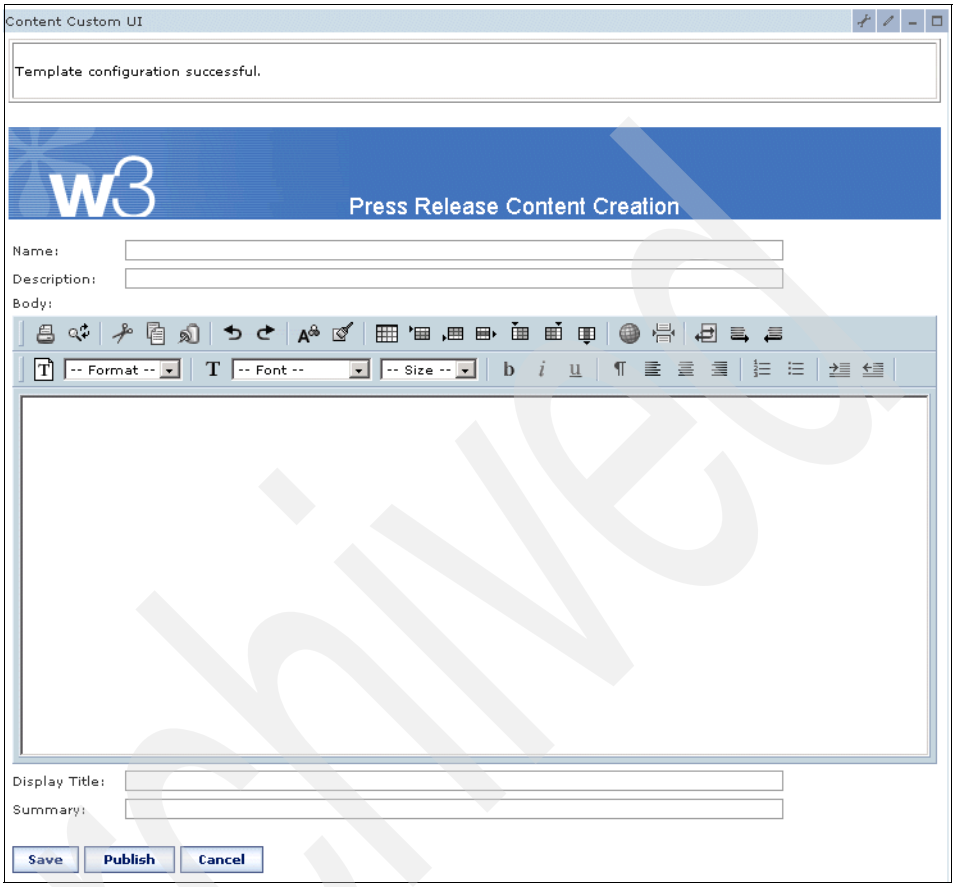


Figure 13-10 Custom Content UI after configuration

4. Review the Content Custom UI portlet. It shows the ID information and the fields of the Bulletin Board template. This information is the only information a user needs to complete the form. Pressing Cancel on the Content Custom UI refreshes the portlet and removes the message at the top.

IBM Workplace Web Content Management sample Workflow Portlet configuration

The Content Custom UI and the Workplace Web Content Management Sample Workflow Portlets work together to create the Quick Edit authoring environment. So, you need to coordinate the Workflow Portlet settings with the Custom Content UI settings (Figure 13-11 on page 546).

Archived

Figure 13-11 Workflow Portlet settings

To coordinate the settings, follow these steps:

1. Go to Configuration of the Workplace Web Content Management Sample Workflow Portlet.
2. Select the Bulletin Board authoring template to show a list of items that are in draft, published, or expired status for that template. The Target Page Mapping for Preview field allows you to set a page for preview if it is other than the current page. Because the preview is on the same page as the other Portlets, you can keep the Target Page Mapping for Preview page empty.
3. Select **Submit**.

The configuration is now complete and the new Quick Edit authoring page Department News is ready for use (Figure 13-12 on page 547).

Archived

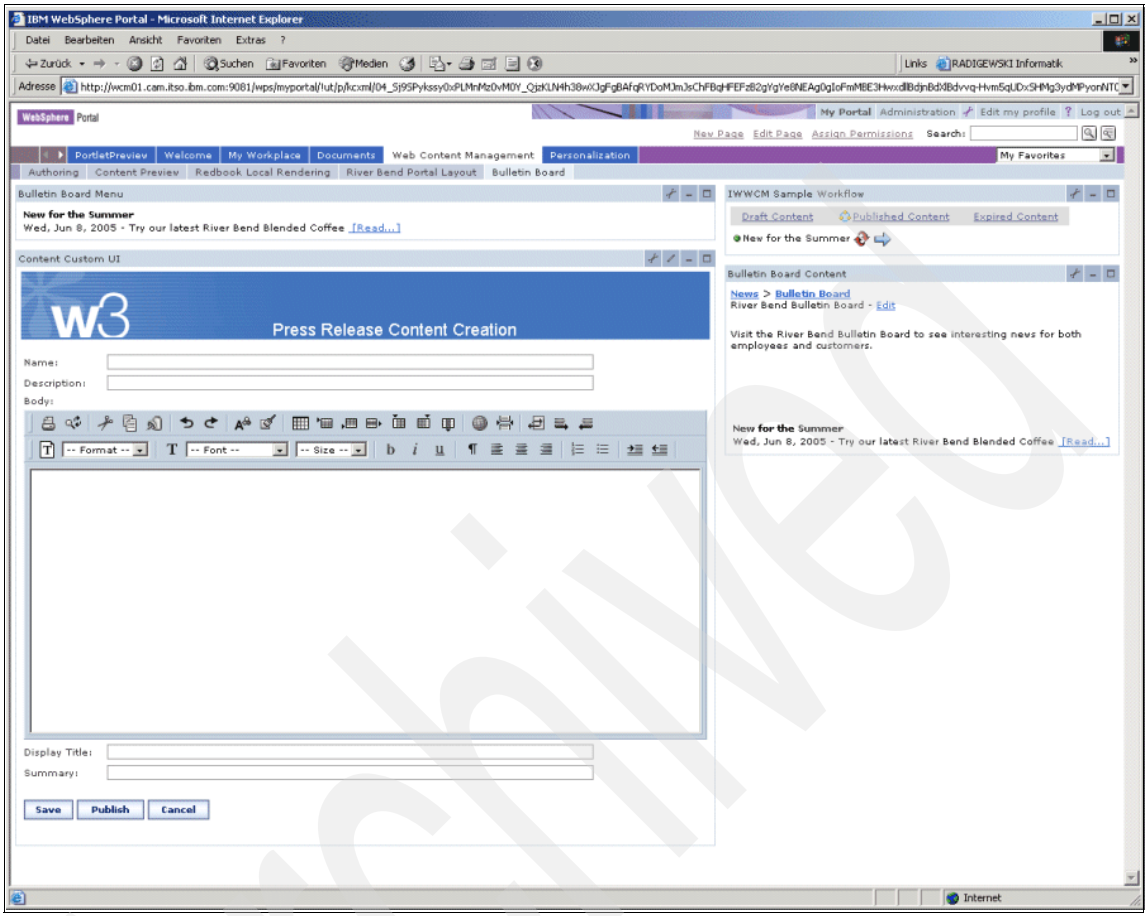


Figure 13-12 Configured Bulletin Board contribution page

Archived

13.2.4 Demonstration of Quick Edit

To demonstrate how easily you can create content, follow these steps:

1. Enter content attributes into form.

   In the Content Custom UI Portlet, complete the fields for the new Bulletin Board article (Figure 13-13). Be sure to include name and title fields.

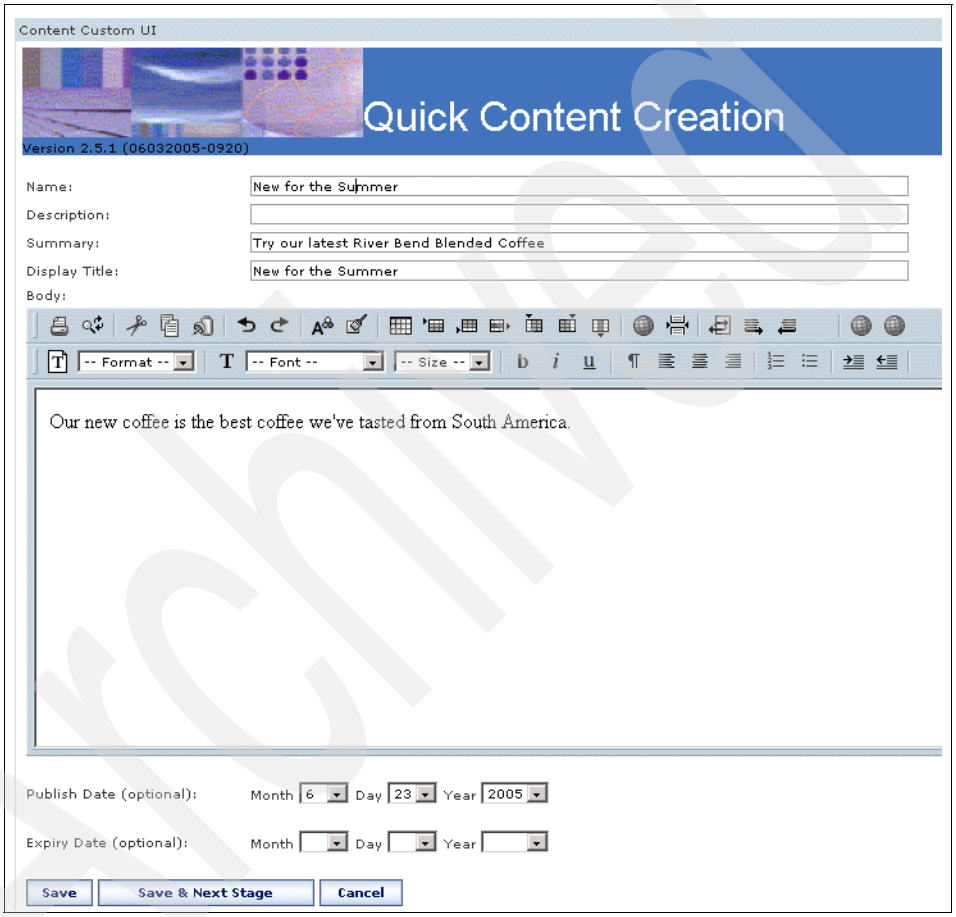


Figure 13-13 Complete content for Bulletin content

Note that you can set the effective and expiry dates. These dates can be left blank to take the default, which is the publish date (today) or the effective date, with no planned expiration.

Archived

The new news item now appears as a draft item in the Workplace Web Content Management Sample Workflow Portlet (Figure 13-14).

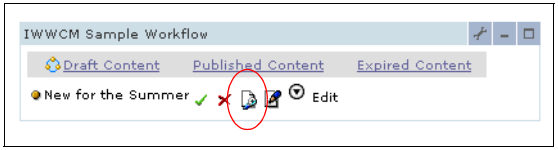


Figure 13-14 Preview the content

2. Preview content as it would appear on Web site.

   If you select Preview (identified with hover help) in the Workflow list, you should see the item in the Department News Detail portlet below it (Figure 13-15).

Figure 13-15 Preview the content

   If you need to make any changes, you can select the Edit icon or Click-to-Action function to populate the Custom Content UI Portlet with the News Article values. You can make changes easily and then submit the content. You can repeat this action until the content is ready to be published.

3. Publish the content to the Web site.

   When you are satisfied with the entry, select Publish (green check mark) in the Workplace Web Content Management Sample Workflow Portlet. The Department News Detail Portlet shows the item as published on the live site.

   The Workplace Web Content Management Sample Workflow Portlet moves the content to the next stage when Approve is selected. Because the Quick Edit Workflow next stage is Publish, the content is published. If there are more stages before Publish, all draft stages need to be approved before the content is published to the Web site.

Archived

13.2.5 Customizing the Content Custom UI Portlet

This section examines the default JSPs as well as two custom JSPs that show how the Content Custom UI Portlet can produce a customized authoring environment.

The JSP files that correspond to this Portlet are stored in the following directory:

<*Drive*>:\<*Path*>\WebSphere\PortalServer\installedApps\CustomUI_PA_1_0_P1.ear\PA_1_0_P1.war\wcm_customui\jsp\html

Additional JSP examples from the Sampleset.zip are stored in the additional subdirectory, *sampleset1* under *html*.

The presentation JSP file CustomUIPortletFormView.jsp is the default JSP. You can change this default in the Configure and Edit modes of the Portlet, or you can modify the default to your own format. For example, you can change the image to your company logo.

It is recommended that before you change any JSP files or updating or reinstalling the Portlet, you copy JSP files and any other image or JavaScript to a back-up directory outside of the Portal.

Also, note that the Workplace Web Content Management Sample Workflow JSP files are found in the following directory:

<*Drive*>:\<*Path*>\WebSphere\PortalServer\installedApps\iwwcm_samp_kflow_list_PA_1_0_P2.ear\PA_1_0_P2.war\IWWCMWorkflow\jsp\htm

Default JSP

The default JSP file CustomUIPortletFormView.jsp for the Content Custom UI Portlet has three sections: header, body, and footer. The header and footer are separated in specific JSP files to allow them to be shared easily in the different main JSP files.

The header, inserted from CustomUIPortletViewHeader.JSP, includes all the needed tag libraries and initializes the session bean and the workspace (Example 13-4 and Example 13-5 on page 551).

Example 13-4 CustomUIPortletViewHeader.JSP - Abstract

```
<%@ page session="false" contentType="text/html" import="java.util.*,
com.ibm.workplace.wcm.customui.portlet.*,
com.ibm.workplace.wcm.customui.portlet.util.*, com.ibm.workplace.wcm.api.*"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%-- Load Portal tag library and initialize constants --%>
<%@ taglib uri="/WEB-INF/tld/engine.tld" prefix="wps" %>
```

```
<%-- Load the Rich Text Editor Tag Library --%>
<%@ taglib uri="/WEB-INF/tld/DocEditor.tld" prefix="rte" %>
```

Example 13-5 CustomUIPortletViewHeader.JSP - Abstract

```
<%
   CustomUIPortletViewBean viewBean =
(CustomUIPortletViewBean)portletRequest.getAttribute(CustomUIPortlet.VIEW_BEAN)
;
   CustomUIPortletSessionBean sessionBean =
(CustomUIPortletSessionBean)portletRequest.getPortletSession().getAttribute(Cus
tomUIPortlet.SESSION_BEAN);
    Vector rteFields = new Vector();
    Workspace workspace = sessionBean.getWorkspace();
    // This helps the wcm tag env use the same workspace
    pageContext.setAttribute(Workspace.WCM_WORKSPACE_KEY, workspace);
    %>
<wcm:initworkspace user="<%= (java.security.Principal)portletRequest.getUser()
%>" />
```

The body of the JSP file is contained in the default JSP itself and contains an iterator that cycles through the template fields and handles the display of the fields according to the field type (Example 13-6).

Example 13-6 Template fields iterator

```
Map templateFields = sessionBean.getTemplateFields();
for (Iterator e = templateFields.keySet().iterator(); e.hasNext();)
{
   String fieldName = (String) e.next();
```

Each field gets the attribute name as the label (Example 13-7).

Example 13-7 Template labels

```
<LABEL class="wpsLabelText"><%=fieldName%>:</LABEL>
<%String fieldClass = (String) templateFields.get(fieldName);
```

Archived

For text fields, a string entry field is created (Example 13-8). Note that the name of the input field is created using the `Namespace` appended with the `Field` name. The value of the input field is retrieved from the session bean referenced by the field name.

Example 13-8 Template fields

```
if (fieldClass.equals("com.ibm.workplace.wcm.api.WCM_TextComponent"))
{%>
<td> <INPUT class="wpsEditField"
name="<portletAPI:encodeNamespace value='<%=fieldName%>'/>"
type="text" size ="100%"
value="<%=sessionBean.getContentValues(fieldName)%>"/></td>
```

For Rich Text fields, an IBM Rich Text editor is used (Example 13-9).

Example 13-9 Rich Text Editor

```
if (fieldClass.equals("com.ibm.workplace.wcm.api.WCM_RichTextComponent") ||
fieldClass.equals("com.ibm.workplace.wcm.api.WCM_HTMLComponent"))
{String fieldNameWithNameSpace=portletResponse.encodeNamespace(fieldName);
rteFields.addElement(fieldNameWithNameSpace);
String editorName = fieldNameWithNameSpace + "Editor";
   editorName = editorName.replace(' ' , '_');
   String listener = "IBM_RTE_setEditorHTML('" + editorName + "', '" +
sessionBean.getContentValues(fieldName)+ "')";
```

Image fields get a file selection field (Example 13-10).

Example 13-10 Image fields

```
if (fieldClass.equals("com.ibm.workplace.wcm.api.WCM_ImageComponent") ||
fieldClass.equals("com.ibm.workplace.wcm.api.WCM_FileComponent"))
{%>
<td><INPUT dir="<%=bidiFileLTR%>" class="wpsEditField"
name="<portletAPI:encodeNamespace value='<%=fieldName%>'/>" size="100%"
type="file"/></td>
```

Other fields types are not given entry fields, just labels. The footer performs the fields validation and handles the submit, publish, and cancel actions.

Example 13-11 JavaScript actions

```
function <portletAPI:encodeNamespace value='CustomUI'/>setPublish()
function <portletAPI:encodeNamespace value='CustomUI'/>setCancel()
function <portletAPI:encodeNamespace value='CustomUI'/>doSubmit()
function <portletAPI:encodeNamespace value='CustomUI'/>validateFields()
```

Attribute formatting

This section explains the formatting for attributes.

Using categories and keywords

When in the edit mode of the Customizable Template Portlet, you can select whether categories and publish and expiry dates are displayed (Figure 13-16). When the settings are selected, the default JSP renders multi-selection boxes for categories, shows text areas for keywords, and displays the date selection values (Figure 13-17). By default, the setting of these dates is optional.

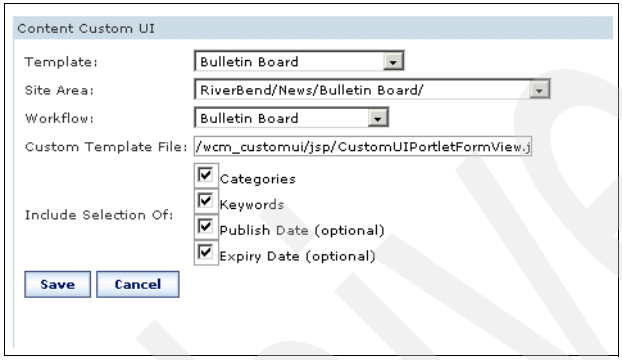


Figure 13-16 Edit mode settings

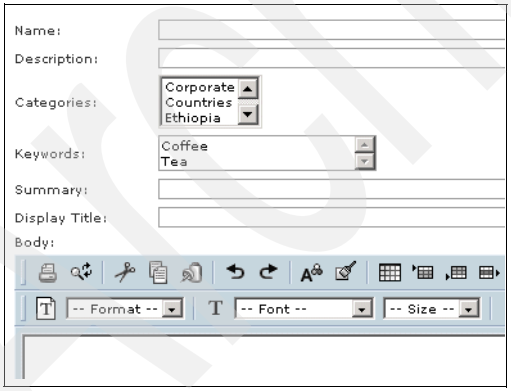


Figure 13-17 Category and keyword selection fields

Setting dates

You can set the effective and expiry dates explicitly (Figure 13-18 on page 554). If these settings are left blank, the default is that the publish date (today) is equal to the effective date, and there is no planned expiration. Sample JavaScript is provided (in the CustomUIPortletViewFooter.jsp) to perform some validation on

Archived

the effective and expiry dates. Each date contains three input fields (month, day, and year). If one of these input fields is set, all three must be set. If invalid dates are selected, a message alerts the user that the date is not set correctly.

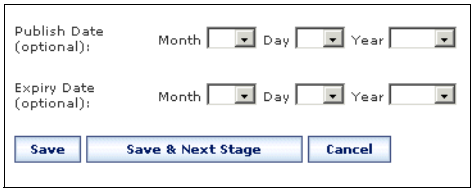


Figure 13-18 Publish and expiry date fields

Sample Edits

JavaScript is provided to validate the name that the user has entered for a content. The code checks to ensure that the content name does not contain any invalid characters (/, \, % and ?). If the content name contains an invalid character, a message alerts the user that the content name is invalid.

Sample Rich Text editor extensions

When creating a new content item using the Bulletin Board template, an author might want to create a link to another content item such as a previously published article or an image (or other Library component). IBM Workplace Web Content Management allows `AptrixCmpnt` tags to be inserted into a Rich Text field that can be formatted to provide this capability. Two new icons are added to the Rich Text editor to create these links — one for link referencing an existing content item and the other for a library object (Figure 13-19). Hover help indicates each icon's function.

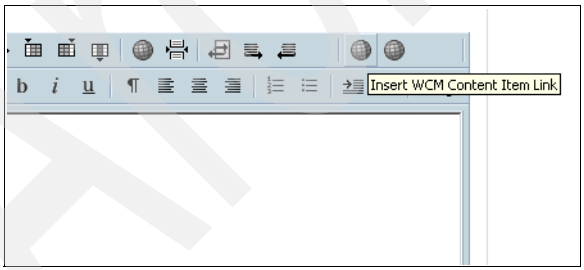


Figure 13-19 Sample button on Rich Text editor to insert a link to a content item

Adding Links to other Content

If you click the Insert Content Item icon, a window displays that filters the content items and allows you to select a content item for the reference link (Figure 13-20 on page 555).

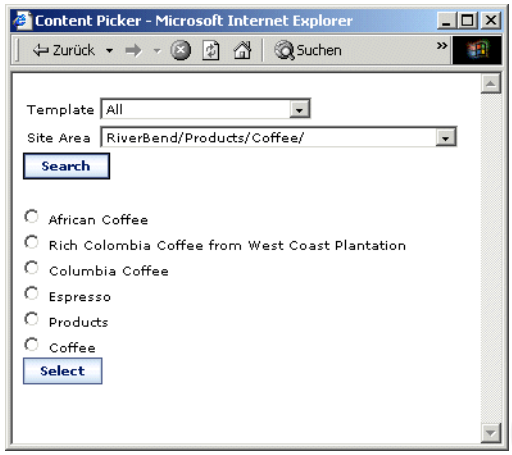


Figure 13-20 Sample content item selection

When selected, the correct Aptrix Component tag is inserted into the Rich Text editor field.

Important: There is no visual indication for the link until the Rich Text editor field is used in a presentation template. If you select the scroll in the upper-right of the Rich Text editor, the HTML is displayed. When the item is saved and previewed, the link is rendered.

Archived

Adding Link to Library items

You can add a link to a Library item in a similar manner. Do the following:

1. Select the adjacent icon (Insert Library Component Link)
2. Select a library component. The resulting tag is generated (Figure 13-21).

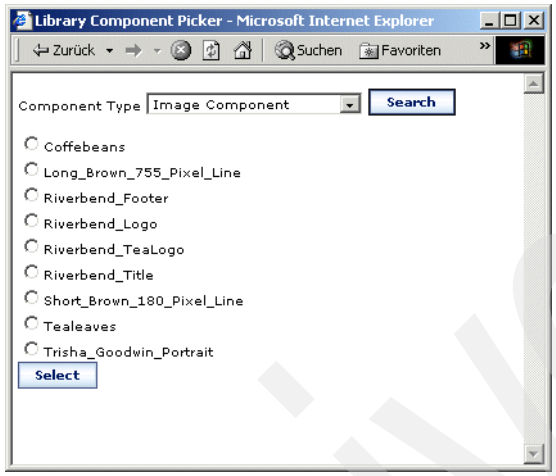


Figure 13-21 Sample selection of an Image library component

As was the case when inserting a content item, you can see the generated HTML for your link (Figure 13-22) and see the embedded library component itself when you preview the HTML (Figure 13-23 on page 557).

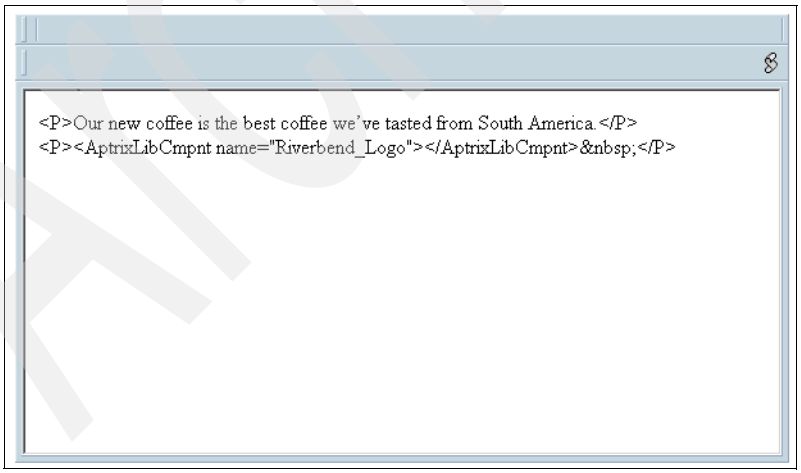


Figure 13-22 Generated HTML for link

Archived

Figure 13-23 Image preview

Customizing a JSP format

To create a custom JSP to use with the Customizable Template Portlet rather than the default, simply make a copy of the default and add your custom JSP code.

As an example, use the CustomUIPortletFormViewNewsButton.jsp that is provided with this book. Unzip the CustomJSP.zip file into the following directory:

<*Drive*>:\<*Path*>\WebSphere\PortalServer\installedApps\CustomUI_PA_1_0_1BP.ear\PA_1_0_1BP.war\wcm_customui\jsp\html

This custom JSP requires proper placement of the following additional files:

- Place the welcome_5.gif file in the images directory under the html directory.
- Unzip the utils.zip at the same level as the wcm_customui\jsp directory.

In this JSP, a new icon takes the place of the default. Each field of the template is aligned in an HTML table to allow all fields to fit on the browser window without the user having to scroll. Use the Edit icon to change the JSP name in the Custom Template File field.

Archived

Table 13-1 lists the JSPs that the CustomUIPortletFormViewNewsButton.jsp file includes.

Table 13-1 JSPs included with CustomUIPortletFormViewNewsButton.jsp

Included JSP	Functions
CustomUIPortletViewHeader.jsp	Taglibs and session object
utils/chooser.jsp	Sample pop-up selection to populate Title field
CustomUIPortletViewImageTable.jsp	Image reference
CustomUIPortletViewRequiredFields.jsp	Name and Description fields
CustomUIPortletViewCategories.jsp	Category and Keyword entry fields
CustomUIPortletViewDates.jsp	Publish and expiry date entry fields
CustomUIPortletViewButtons.jsp	Action buttons (Save, Save & Next Stage, Cancel)
CustomUIPortletViewFooter.jsp	Field validation and submit

Field Names

You must code the name and description fields to add a content item to the repository (Example 13-12).

Example 13-12 Field names

```
<%-- The Name and Description field MUST exist --%>
<tr> <td> <LABEL class="wpsLabelText">
<portletAPI:text
bundle="com.ibm.workplace.wcm.customui.portlet.customPortletUI"
   key="NAME" />:</LABEL></td>
<td> <INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=UIConstants.NAME%>'/>" type="text" size="100%"
value="<%=sessionBean.getContentValues(UIConstants.NAME)%>" tabindex="1"/></td>
          </tr>

<tr> <td> <LABEL class="wpsLabelText">
<portletAPI:text
bundle="com.ibm.workplace.wcm.customui.portlet.customPortletUI"
key="DESCRIPTION" />:</LABEL></td>
<td> <INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=UIConstants.DESCRIPTION%>'/>" type="text" size="100%"
value="<%=sessionBean.getContentValues(UIConstants.DESCRIPTION)%>"
tabindex="2"/></td>
</tr>
<%-- End required fields --%>
```

Then, for each template field, use the name from the authoring template component name that is appended to the `Namespace` as the field name. Get the value of the field from the session as shown in Example 13-13. If you use these conventions, you will be able to render with any HTML input field.

Example 13-13 Getting value for the template field

```
<td class="ArticlePagesBodycopyBold" width="151">News Title:</td>
<td width="1176"><TEXTAREA rows="2" cols="100" class="wpsEditField"
name="<portletAPI:encodeNamespace
value='NewsTitle'/>"><%=sessionBean.getContentValues("NewsTitle")%></TEXTAREA>
```

Adding a button

You can add functionality to the JSP as with any other JSP Portlet application. In Example 13-14, a button is added to show a selection list to fill in the title. The JSPs referenced are included in the utils.zip file and should be unzipped into the following directory:

<Drive>:*<Path>*\WebSphere\PortalServer\installedApps\CustomUI_PA_1_0_S5.ear\PA_1_0_S5.war\wcm_customui

Example 13-14 Adding a button to show a selection list to fill in the title

```
<input type=button value="Search for Title"
onClick="javascript:chooseItem('<%=response.encodeURL("/wcm_customui/utils/News
Title.jsp")%>','<portletAPI:encodeNamespace
value='CustomTemplate'/>','<portletAPI:encodeNamespace
value='NewsTitle'/>')"></td>
   </tr>
```

When the button is selected the NewsTitle.jsp displays a selection of titles. When a selection is made, the title input field is populated (Figure 13-24). The list of titles is hard coded but could be supplied from any query results.

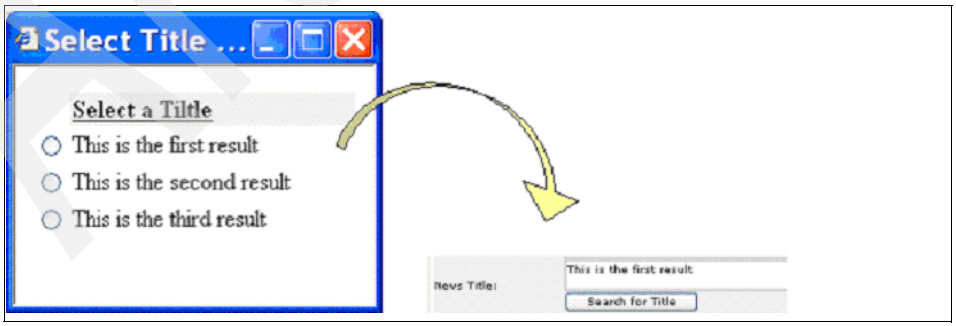


Figure 13-24 Title selection and field population

Using multiple templates

Multiple templates can also use the same Content Portlet. This action is allowed by overriding the template, site area, and workflow settings. See the CustomUIPortletFormView_t1.jsp in the Sampleset.zip file which is deployed in the following directory:

> *<Drive>*:*<Path>*\WebSphere\PortalServer\installedApps\CustomUI_PA_1_0_S5.ear\PA_1_0_S5.war\wcm_customui\jsp\html\sampleset1

To use this JSP file in your Content Portlet, configure it with sampleset1\CustomUIPortletFormView_t1.jsp in the Custom Template File field.

The JSP has two divisions — one for each template format. Both templates contain fields 1 through 4. In Example 13-15, the MyTemplate1 requires fields 2 and 4 and MyTemplate2 requires fields 1 and 3.

Example 13-15 JSP field data for each template format

```
<div id="<portletAPI:encodeNamespace value='MyTemplate1'/>" style="display: <%=
currentTemplate.equals("MyTemplate1")?"block":"none"%>">
<table>  <tr>
<td> <LABEL class="wpsLabelText"><%=t_field2%>:</LABEL></td>
<td> <INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=t_field2%>'/>" type="text" size ="50%"
value="<%=sessionBean.getContentValues(t_field2)%>"/></td>
</tr><tr><td> <LABEL class="wpsLabelText"><%=t_field4%>:</LABEL></td>
<td> <INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=t_field4%>'/>" type="text" size ="50%"
value="<%=sessionBean.getContentValues(t_field4)%>"/></td>
</tr></table>
</div></td></tr>

...

<div id="<portletAPI:encodeNamespace value='MyTemplate2'/>" style="display: <%=
currentTemplate.equals("MyTemplate2")?"block":"none"%>">
<table> <tr>
<td nowrap> <LABEL class="wpsLabelText"><%=t_field1%>:</LABEL>
<INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=t_field1%>'/>" type="text" size ="50%"
value="<%=sessionBean.getContentValues(t_field1)%>"/></td>
   <td nowrap> <LABEL class="wpsLabelText"><%=t_field3%>:</LABEL>
<INPUT class="wpsEditField"  name="<portletAPI:encodeNamespace
value='<%=t_field3%>'/>" type="text" size ="50%"
value="<%=sessionBean.getContentValues(t_field3)%>"/></td>
 </tr></table></div>
```

The content contributor can the use the same Portlet for either template.

13.2.6 Customizing the Workplace Web Content Management Sample Workflow Portlet

The Workplace Web Content Management Sample Workflow JSP file has three divisions that are used to render draft, published, and expired content. This Portlet uses inter-portlet communications to interact with the Customizable Template Portlet and Preview Portlet. The workflow JSP file IWWCMWorkflowPortletView.jsp sends messages in the following formats to the CTP:

```
"action = Action"
"Attribute = Value"
```

So, to send the message to the Customizable Template Portlet for editing the content, you would need to send two messages, as follows:

```
action=EDIT
docId=selectedDocId
```

The following code generates this message:

```
<a href='<portletAPI:createURI><portletAPI:URIAction
name='<%=IWWCMWorkflowPortlet.WORKFLOW_ACTION%>'/><portletAPI:URIParameter
name="<%=IWWCMWorkflowPortlet.DOCID%>"
value="<%=docId%>"/><portletAPI:URIParameter
name="<%=IWWCMWorkflowPortlet.ACTION%>"
value="<%=IWWCMWorkflowPortlet.EDIT%>"/></portletAPI:createURI>'><img
style="vertical-align:middle;border=0"src="<%=request.getContextPath()%>/IW
WCMWorkflow/jsp/html/images/ajpe_btnicon_edit.gif" alt="Edit"/></a>
```

The Preview Portlet receives messages in the following format:

```
WCM_GLOBAL_CONTEXT=[site area]/[content name]?id=[value}
```

The following code formats this message:

```
<a href="<%=currentURI %>
?WCM_GLOBAL_CONTEXT=<%=IWWCMWorkflowPortlet.getContentUrl(docId,portletRequ
est,sessionBean)%>">
<img style="vertical-align:middle;border=0"
src="<%=request.getContextPath()%>/IWWCMWorkflow/jsp/html/images/ajpe_btnic
on_preview.gif" alt="Preview" /></a>
<%} %>
```

13.3 Extensions to the Customizable Template Portlet

This section describes the source code that implements the Link Management Extensions that are discussed in 9.5, “Implementing a custom display Portlet” on page 418. Refer to this section for a detailed explanation of the concept behind these extensions.

For the user interaction the Customizable Template Portlet implements an ActionListener (`actionPerformed` method). This method is also responsible for the save event. It delegates the write to IBM Workplace Web Content Management backend operation to a method called `writeToWCM()`. To implement the link management, you have to extend this method.

After the content document is saved [`workspace.save(content)`], you add some code that creates the link-list. As outlined in Chapter 9, “Advanced integration with WebSphere Portal” on page 403, we have to parse the rendered content for all occurrences of the `link2content` JavaScript function.

Example 13-16 renders the content document and then passes the resulting HTML to the `parseContent` method.

Example 13-16 Rendering and passing HTML to the parseContent method

```
try {
   RenderingContext context = workspace.createRenderingContext(p_request,
null, map);
   context.setRenderedContent("/WP Experts/Home/" + content.getName());
   String rendered = workspace.render(context);
   System.err.println("content rendered: " + rendered);
   parseContent(rendered, content);
   p_sessionBean.setContentSaveResults(workspace.save(content));
   p_sessionBean.setContentSaveResults(workspace.save(siteArea));
} catch (Exception ge) {
   ge.printStackTrace();
}
```

The parseContent method then uses a regular expression to build the link list, as shown in Example 13-17.

Example 13-17 parseContent method to build a link list

```
try {
   Pattern Regex =
Pattern.compile("(?:link2content\\(')(.*)(?:',')(.*)(?:'\\))",
Pattern.CANON_EQ);
   Matcher RegexMatcher = Regex.matcher(content);
   while (RegexMatcher.find()) {
      for (int i = 1; i <= RegexMatcher.groupCount(); i++) {
```

```
            sb.append(RegexMatcher.group(i));
            sb.append(";");
         }
      }
      sb.deleteCharAt(sb.length() - 1);
   } catch (PatternSyntaxException ex) {
      // Syntax error in the regular expression
   }
```

After this the result is stored in the links field, as shown in Example 13-18.

Example 13-18 Result is stored in the links field

```
if (doc.hasComponent("links")){
   TextComponent bodyComponent;
   try {
      System.err.println("setting component");
      bodyComponent = (TextComponent) doc.getComponent("links");
      bodyComponent.setText(sb.toString());
      doc.setComponent("links", bodyComponent);
   } catch (ComponentNotFoundException e) {
      e.printStackTrace();
   } catch (IllegalTypeChangeException e) {
      e.printStackTrace();
   }
}
```

In the next step, you use the built link list in the rendering phase of the Custom Workplace Web Content Management Portlet. You do this in the `doView` method of the Portlet. As outlined in Chapter 9, “Advanced integration with WebSphere Portal” on page 403, you have to create the final links in this step. 9.5, “Implementing a custom display Portlet” on page 418 explains the concepts behind this.

The code builds a JavaScript array for us, as shown in Example 13-19. The result is JavaScript code that you use in a later step. It uses a `StringTokenizer` to cut the link list into pieces and then calls `getPortalLink` to create the links. The resulting JavaScript array uses the combination of `pagename+docId` as its key.

Example 13-19 JavaScript which uses a StringTokenizer to cut the link list into pieces

```
TextComponent bodyComponent = (TextComponent) doc.getComponent("links");
StringTokenizer stok = new StringTokenizer(bodyComponent.getText(), ";");
linkBuf.append("var links = new Array();\n");
while (stok.hasMoreElements()) {
   String pageName = stok.nextToken();
   if (stok.hasMoreElements()) {
      String docId = stok.nextToken();
```

```
        linkBuf.append("links[\"");
        linkBuf.append(pageName);
        linkBuf.append(docId);
        linkBuf.append("\"] = \"");
        linkBuf.append(getPortalLink(request, pageName, docId));
        linkBuf.append("\";\n");
    }
}
```

The `getPortalLink` method makes use of the `CreateURLCommand` class to create links to a Portal page, as shown in Example 13-20. It also allows you to add a URL parameter.

Example 13-20 The getPortalLink method makes use of the CreateURLCommand

```
try {
   CreateUrlCommand iCreateUrlCommand = new CreateUrlCommand();
   iCreateUrlCommand.setRunData(runData);
   iCreateUrlCommand.setContentNode(iComposition);
   iCreateUrlCommand.setPacCheck(null);
   iCreateUrlCommand.setRequestID(true);
   iCreateUrlCommand.addQueryData(DOCUMENT_ID, param);
   iCreateUrlCommand.execute();
   URL iURL = iCreateUrlCommand.getURL();
   if (iURL != null) {
      uri = iURL.toString();
   }
} catch (CommandException e) {
   System.err.println(e.getMessage());
}
```

Now, you can make use of the JavaScript array. Example 13-21 embeds the link list array on the PortletView.jsp and also implements the `link2content` function. The function takes the unique name of the Portal page and the document ID as input. It then does a lookup in the JavaScript array and resolves the link. Finally, the link then is executed.

Example 13-21 JavaScript array embedding the link list array

```
<script language="JavaScript" type="text/javascript">
<!--
<%= portletRequest.getAttribute(WCMCustomPortlet.JS_LINK_ARRAY) %>
function link2content(node, id) {
   window.location.href=links[node+id];
}
//-->
</script>
```

14

Migrating to IBM Workplace Web Content Management 5.1

This chapter describes how to migrate a Web site from a previous version of IBM Workplace Web Content Management into IBM Workplace Web Content Management 5.1. It includes the following sections:

- Overview of migrating data from a previous version
- Transferring file system data
- Verifying and troubleshooting file system data transfer
- Migrating users and groups
- Verifying and troubleshooting user and group migration
- Creating the unrestricted resources
- Migrating data from a previous version
- Enabling user migration
- Migrating users directly to WebSphere Member Manager
- Migrating users to WebSphere Member Manager using a user migration mapping file
- Disabling user migration

© Copyright IBM Corp. 2006. All rights reserved.

14.1 Overview of migrating data from a previous version

In earlier versions of Lotus Workplace Web Content Management, you could store data and users in the file system as well as in the supported JDBC databases and IBM Content Manager. In IBM Workplace Web Content Management 2.5 and 5.1, the file system is no longer a supported data persistency.

If you upgrade from an older version of Lotus Workplace Web Content Management and your data was stored on the file system, you need to follow the correct data transfer and migration procedures. If your data was stored in a JDBC database or IBM Content Manager, you do not need to transfer the data, but you do need to configure IBM Workplace Web Content Management to access the data. For all data storage types, you need to migrate the Lotus Workplace Web Content Management users and groups to IBM WebSphere Member Manager and configure the data so that it uses the WebSphere Member Manager users and groups.

Note: You can upgrade from any previous version of Lotus Workplace Web Content Management. To upgrade from a version of Aptrix, first upgrade to one of the versions of Lotus Workplace Web Content Management and then upgrade to IBM Workplace Web Content Management 2.5 or 5.1. The steps for upgrading Aptrix to Lotus Workplace Web Content Management are out of the scope of this book. Refer to the Lotus Workplace Web Content Management Information Center for the steps for upgrading Aptrix at:

http://www-128.ibm.com/developerworks/workplace/documentation/webcontentmanagement/

Note: IBM Workplace Web Content Management 2.5 requires WebSphere Portal 5.0.2. If you upgrade to Portal 5.1, you need to upgrade to IBM Workplace Web Content Management 5.1. When upgrading from IBM Workplace Web Content Management 2.5 to IBM Workplace Web Content Management 5.1, you need to uninstall 2.5 and install 5.1 on Portal 5.1. There is no need to migrate the IBM Workplace Web Content Management data, because version 2.5 stores data in a supported repository and uses a supported user repository such as WebSphere Member Manager or LDAP. This book describes the process for upgrading from a previous version of Lotus Workplace Web Content Management to IBM Workplace Web Content Management 2.5 or 5.1.

14.1.1 Differentiating upgrading, transferring, and migrating

The terms *upgrading*, *transferring*, and *migrating* are used when describing the process of moving from a previous version of Lotus Workplace Web Content Management to a newer version.

Upgrading refers to the process of installing a newer version of IBM Workplace Web Content Management. This process includes:

- Transferring, which refers to the process of moving existing data from one repository to another.
- Migrating, which refers to the process of recreating existing Lotus Workplace Web Content Management users and groups in WebSphere Member Manager and changing the Workplace Web Content Management data so that it points to the new WebSphere Member Manager users and groups.

14.1.2 Upgrading IBM Workplace Web Content Management

Table 14-1 provides an overview of the steps for upgrading IBM Workplace Web Content Management. Figure 14-1 on page 568 illustrates these steps.

Table 14-1 Steps for upgrading Workplace Web Content Management

Step	Action
1	Back up your existing data repository and users folder prior to upgrading as follows: ► Back up your existing data using the backup solution recommended for the data repository. ► Back up your existing users by copying the [ILWWCM_HOME]\connect\users folder. This folder does not contain groups, which are saved in the data folder. The users are also saved in the data folder. **Note:** IBM Workplace Web Content Management does not have a built-in backup feature.
2	Install the new version of IBM Workplace Web Content Management.
3	Configure the new data repository. **Note:** If you intend to use IBM Cloudscape database, there is no need to perform this step, because at installation IBM Workplace Web Content Management configures Cloudscape as the default data repository.
4	Edit the wpconfig.properties file to point to the file system location. **Note:** This step is only required if the existing data is stored in the file system.

Archived

Step	Action
5	Transfer existing data into the new data repository. **Note:** This step is only required if the existing data is stored in the file system.
6	Migrate the existing Lotus Workplace Web Content Management users and groups to WebSphere Member Manager and configure the data to use the new WebSphere Member Manager users and groups.
7	Run the resource checker utility to recreate the resources in the *WEB_APP_HOME* folder so that the unrestricted resources appear on the Web site.
8	Test your upgraded data and correct any errors using the troubleshooting tips.

Important: Test the upgrade in a test environment before upgrading the production environment.

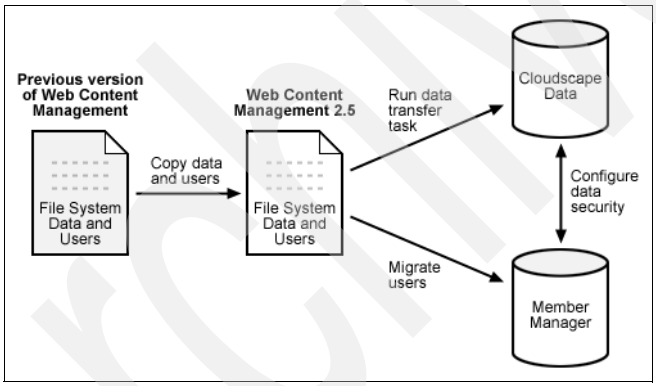


Figure 14-1 Transferring file system data and migrating users

14.2 Transferring file system data

IBM Workplace Web Content Management 2.5 and 5.1 can use data from an older version of Lotus Workplace Web Content Management. The only changes that might be required when upgrading are as follows:

- If the older version stored data in the file system, then the data needs to be transferred to a Cloudscape database and then, if required, to one of the supported data repositories, which include a JDBC database or Content Manager.

- If the older version stored data in a supported JDBC database or Content Manager, then you do not need to do anything to the data unless you plan to use a different JDBC database.

14.2.1 Transferring file system data

Table 14-2 provides an overview of the steps for transferring file system data from an older version of Lotus Workplace Web Content Management.

Table 14-2 Transferring file system data

Step	Action
1	Copy the existing Lotus Workplace Web Content Management data folder to the *ILWWCM_HOME*\ilwwcm folder. **Notes:** ► The file system data folder of the previous version is located in the previous version's *ILWWCM_HOME*\ilwwcm folder. ► The destination folder can be different, provided that the path specified in the wpconfig.properties file is correct.
2	Point to the copied file system data by entering the full path to the data folder in the wpconfig.properties file. The following is an example of how the line of code might appear: `WcmFileSystemDataLocation=C:/WebSphere/PortalServer/wcm/ilwwcm/data`
3	Run the transfer-wcm-filesystem-to-cloudscape task to transfer the file system data from the *ILWWCM_HOME*\ilwwcm folder to a Cloudscape database.
4	If required, transfer the data to another database type. **Note:** If you intend to store data in another database type, you must first transfer the file system data to a Cloudscape database and then transfer it to the required database type.

Note: IBM Workplace Web Content Management must be installed before you can transfer the file system data to a Cloudscape database.

Archived

14.2.2 Using the wpconfig.properties file during data transfer

The system uses the following parameter to locate the Lotus Workplace Web Content Management data on the file system:

```
WcmFileSystemDataLocation
```

The wpconfig.properties file is located in the *WPS_HOME*\config folder.

14.2.3 Running the data transfer task

Run the task from the *WPS_HOME*\config directory using a command prompt as follows:

- In Windows, at the command prompt type the following:

  ```
  wpsconfig.bat transfer-wcm-filesystem-to-cloudscape
  ```

- In UNIX, at the command prompt type the following:

  ```
  wpsconfig.bat/.sh transfer-wcm-filesystem-to-cloudscape
  ```

14.2.4 Ensuring valid data format

If you plan to use an IBM DB2 database on z/OS®, make sure the data that was stored in the file system has a valid format. If the format is invalid, problems occur when you transfer the data from the Cloudscape database to the DB2 database on z/OS.

An XML file format is considered valid only if it has one of the following end of line styles:

- Carriage return and line feed (CRLF), which is the Windows end of line style
- Line feed (LF), which is the UNIX end of line style

An XML file that contains *both* CRLF and LF is considered *invalid* because it has inconsistent end of line styles.

Archived

14.3 Verifying and troubleshooting file system data transfer

This section presents information that guides you in verifying that the data transfer was a success and in identifying the cause of errors.

14.3.1 Verifying the data transfer

To verify that data transfer from the file system to the Cloudscape database was successful, the following messages should appear:

- `Build Successful` should appear in the command prompt.
- `Utility Completed` should appear in the *WPS_HOME*\log\wcm_migration.log file.

14.3.2 Examining the error messages

If errors occur during file system data transfer, the error messages appear in the command prompt and in the wcm_migration.log file. Three types of errors might occur:

- NullPointerException
- Connection Error: Failed to start database
- No schema exists for the user

NullPointerException

For file system data transfer, a `NullPointerException` indicates that data folders are missing. Do the following to resolve the errors:

- In the wpconfig.properties file, make sure the value of the `WcmFileSystemDataLocation` parameter is correct. This is the path to the copied file system data.
- In the wpconfig.properties file, make sure that the IBM Workplace Web Content Management parameters have values and that the values are correct, particularly the password for the `WcmDbPassword` parameter.

Archived

- Make sure the following data folders exist, particularly the folders indicated in ***italic bold*** font. Create the missing folders, even if they did not exist prior to data transfer. The folders can be empty.
  - [ILWWCM_HOME]\ilwwcm\cmpnts
  - [ILWWCM_HOME]\ilwwcm\contents
  - ***[ILWWCM_HOME]\ilwwcm\directoryProfile***
  - [ILWWCM_HOME]\ilwwcm\itemGatherers
  - [ILWWCM_HOME]\ilwwcm\profileMappers
  - [ILWWCM_HOME]\ilwwcm\resources
  - [ILWWCM_HOME]\ilwwcm\security
  - [ILWWCM_HOME]\ilwwcm\sites
  - [ILWWCM_HOME]\ilwwcm\sites\sitearea
  - [ILWWCM_HOME]\ilwwcm\styles
  - [ILWWCM_HOME]\ilwwcm\subscribers
  - [ILWWCM_HOME]\ilwwcm\syndicators
  - [ILWWCM_HOME]\ilwwcm\taxonomy
  - [ILWWCM_HOME]\ilwwcm\taxonomy\categories
  - [ILWWCM_HOME]\ilwwcm\templates
  - [ILWWCM_HOME]\ilwwcm\workflows
  - [ILWWCM_HOME]\ilwwcm\workflows\actions
  - [ILWWCM_HOME]\ilwwcm\workflows\stages

Connection Error has occurred: Failed to start database

If the build failed and the wcm_migration.log reports the error `Connection Error has occurred: SQL Exception: Failed to start database`, it indicates that the WebSphere Portal server is started. The data transfer can only occur while the WebSphere Portal server is stopped. Stop the WebSphere Portal server by using the **`WebSphere_Portal`** command before running the file system data transfer task.

No Schema Exists

If the build failed and the wcm_migration.log reports an error that no schema exists for the user, it indicates that no database schema has been defined and the database administrator user name values in the wpconfig.properties and aptrixjpe.properties files are different. In the aptrixjpe.properties file, if you do not define a database schema using the `jdbc.tableSchema=[JDBC_SCHEMA]` parameter, then the schema defaults to the user name that is defined by the `jdbc.username=APP` parameter. If you do not define a schema, make sure that the user name values are the same in both files, as follows:

- In wpconfig.properties the default value is `WcmDbUser=APP`.
- In aptrixjpe.properties the default value is `jdbc.username=APP`.

Archived

14.3.3 Known limitations

Table 14-3 describes the known limitations of the IBM Workplace Web Content Management data transfer process.

Table 14-3 Known limitations

Limitation	Description
No directory checking	The data transfer process does not validate that all folders exist. If a folder is missing, a `NullPointerException` error message appears.
No property validation	The data transfer process does not validate the IBM Workplace Web Content Management parameters in the wpconfig.properties file. If a parameter value is missing or is incorrect, a `NullPointerException` error message appears.
No tracing	Because there is no tracing, you cannot debug the errors easily.

14.3.4 Files that are needed by the IBM Support Desk

If you encounter problems that you cannot solve while transferring file system data to a Cloudscape database and if you have a support contract with the IBM Support Desk, send the following files or folders to the support desk:

- wpconfig.properties located in *WPS_HOME*\config
- wcm_migration.log located in *WPS_HOME*\log
- data folder located in *ILWWCM_HOME*\ilwwcm

14.4 Migrating users and groups

WebSphere Member Manager manages the users and groups that are granted access to IBM Workplace Web Content Management items. In previous versions, Lotus Workplace Web Content Management itself created and managed users and groups.

After you transfer the data to the new data repository, you need to migrate the Lotus Workplace Web Content Management users from the previous version to WebSphere Member Manager. When you migrate a Lotus Workplace Web Content Management user or group to WebSphere Member Manager, the migration process changes the references within the IBM Workplace Web Content Management items from the Lotus Workplace Web Content Management user or group to the WebSphere Member Manager user or group

Archived

automatically. The site developer might need to edit selected items whose references were not updated automatically.

Important: When user migration is complete, you need to grant access to the authoring portlet for each user or group.

14.4.1 Migrating users

Table 14-4 provides an overview of the steps for migrating users from an older version of Lotus Workplace Web Content Management.

Table 14-4 Migrating users

Task	Procedure
1	Before uninstalling the previous version of Lotus Workplace Web Content Management, run the User Checker utility with `create=true`. **Notes:** ► The utility ensures that all users in the data repository have a corresponding Lotus Workplace Web Content Management Framework user. ► The Lotus Workplace Web Content Management Framework was the code that created users in previous versions of Lotus Workplace Web Content Management. The Lotus Workplace Web Content Management Framework no longer exists in IBM Workplace Web Content Management 2.5 and 5.1 because its function is performed by WebSphere Member Manager.
2	Back up the existing Lotus Workplace Web Content Management users folder. **Note:** The users folder is located in *ILWWCM_HOME*\connect.
3	Copy the existing Lotus Workplace Web Content Management users folder to the new version's *ILWWCM_HOME*\wcm\connect folder.
4	If you require that WebSphere Portal security be enabled, make sure it is enabled prior to running user migration. By default, WebSphere Portal security is not enabled. **Note:** You can run user migration when WebSphere Portal security is not enabled. If you enable WebSphere Portal security after running user migration, you need to run user migration again.
5	Enable migration by opening the WebSphere Application Server Administrative Console and adding the IBM Workplace Web Content Management migration .jar file (ilwwcm-migration.jar) to the Web Content Management Shared Library (`WcmLib`).

Archived

Task	Procedure
6	Open the Portal migration page and migrate the users and groups using one of the following methods: ► Migrate the users and groups directly to WebSphere Member Manager. ► Generate, review, and then process a user migration mapping file and migrate the users and groups to WebSphere Member Manager using the user migration mapping file. **Note:** The following URL accesses the migration interface: `http://[HOST]:[PORT]/wps/wcm/jsp/migration/migration.jsp`
7	If you use a user migration mapping file, run the **`XMLaccess`** command to create new WebSphere Member Manager users and groups for existing Lotus Workplace Web Content Management users and groups. **Note:** The user migration process specified in Task 6 does not create new users and groups in WebSphere Member Manager. It only compares and verifies existing users and groups.
8	Rebuild the indexes by deleting the system folder that is located in the *ILWWCM_HOME*\ilwwcm folder and start the WebSphere Portal server. **Note:** The system folder contains the indexes.
9	Disable migration by opening the WebSphere Application Server Administrative Console and removing the IBM Workplace Web Content Management migration .jar file (ilwwcm-migration.jar) from the Web Content Management Shared Library (`WcmLib`).

Important: Do not delete the *ILWWCM_HOME*\connect\users folder because you need it to repeat the user migration process. Repeat the user migration process if you enable or disable WebSphere Portal security or if you change your user repository.

Tip: You can run the user migration process when the data is in the Cloudscape database, or later if you transfer the data to another database type. You do not need to run the user migration process every time you transfer data to another database type.

14.4.2 Creating migrated users and groups with WebSphere Member Manager

When you migrate a user to WebSphere Member Manager, it does not recreate a new user if it contains an existing user that is an exact match. It *does* create a new user if the Lotus Workplace Web Content Manager user does not exist in WebSphere Member Manager. WebSphere Member Manager is case-sensitive and does not recognize spaces. Therefore, WebSphere Member Manager creates a new user if the case does not match and creates a new user with underscores (_) in place of spaces if the Lotus Workplace Web Content Management name contains spaces. This behavior also applies to the recreation and migration of groups.

The migration process might result in a user having two user names in WebSphere Member Manager. For example, the user *John Smith* might have a Lotus Workplace Web Content Management user name of *John Smith* and a WebSphere Member Manager user name of *John A Smith*, resulting in the creation of a new WebSphere Member Manager user name to correspond to the existing Lotus Workplace Web Content Management user name.

The migration process might result in a Lotus Workplace Web Content Management user not being created in WebSphere Member Manager. For example, the user *Mary Jones* might refer to a Lotus Workplace Web Content Management user who is not the same *Mary Jones* who is an existing WebSphere Member Manager user. The migration process would cause the IBM Workplace Web Content Management items to reference the existing Mary Jones user in WebSphere Member Manager, resulting in the Lotus Workplace Web Content Management Mary Jones losing access to the Workplace Web Content Management items.

14.4.3 Migrating users and groups

There are two ways that you can migrate Lotus Workplace Web Content Management users and groups to WebSphere Member Manager:

- Migrating directly to WebSphere Member Manager
- Migrating to WebSphere Member Manager using a user migration mapping file

Migrate Lotus Workplace Web Content Management users and groups directly to WebSphere Member Manager if you are sure that existing Lotus Workplace Web Content Management users and groups *exactly* match the existing WebSphere Member Manager users and groups. Direct migration is suitable if you are installing WebSphere Portal for the first time and do not have existing WebSphere Member Manager users or groups.

Migrate Lotus Workplace Web Content Management users and groups using a user migration mapping file if the match is *not* exact between existing Lotus Workplace Web Content Management users and groups and existing WebSphere Member Manager users and groups. A user migration mapping file contains the original Lotus Workplace Web Content Management user and groups and the new WebSphere Member Manager user and groups names. When you generate a user migration mapping file, you can edit the file and map the existing Lotus Workplace Web Content Management users and groups to the correct WebSphere Member Manager users and groups. This method is recommended if you were using WebSphere Portal and WebSphere Member Manager prior to using IBM Workplace Web Content Management.

Important: Direct migration might result in an error message that the required `sn` attribute for each user does not contain a value. The reason for the empty attribute is that users and groups in earlier versions of Lotus Workplace Web Content Management do not provide a `lastname` attribute. If you encounter this error, you need to migrate users and groups using a user migration mapping file and run the **`XMLaccess`** command.

14.4.4 Migrating to WebSphere Member Manager using a user migration mapping file

The migrationUsers.properties file is the user migration mapping file. It contains the mapping of the original Lotus Workplace Web Content Management users and groups to the new WebSphere Member Manager users and groups. You need to review and modify this file before processing it. The original Lotus Workplace Web Content Management users and groups are listed on the left, and the WebSphere Member Manager users and groups are listed on the right. Modify the file as follows:

- Change only the proposed WebSphere Member Manager names.
- Change the property values of the WebSphere Member Manager name to use the distinguished name, for example:

```
uid=username,o=default organization
cn=group name,o=default organization
```

- Do not use an ampersand (&) in user or group names because it causes errors.

The following line of code is an example of how to change the group name `All Staff to All_WP_Staff` using the distinguished name:

```
All\ Staff=cn=All_WP_Staff,ou=groups,dc=ibm,dc=com
```

The migrationUsers.properties file is located in the *ILWWCM_HOME*\migration folder.

14.4.5 Adding new users and groups to WebSphere Member Manager using the migrationUsers.xml file

The user migration mapping file compares only existing WebSphere Member Manager user and group names with existing Lotus Workplace Web Content Management user and group names. When you use the user migration mapping file, WebSphere Member Manager does not create new user or group names if the names are not already in WebSphere Member Manager. In order to create new WebSphere Member Manager users and groups for existing Lotus Workplace Web Content Management users and groups, run the **`XMLaccess`** command.

The **`XMLaccess`** command uses the migrationUsers.xml file for user and group definition. The migrationUsers.xml file contains the required default attributes to define a user or group. If your LDAP server requires different attributes for user or group definition, change the file to contain the appropriate attributes for the LDAP server. Ask your LDAP administrator for the required attributes for your LDAP server.

Important: Users and groups in earlier versions of Lotus Workplace Web Content Management do not provide a `lastname` attribute. You must type a value for the `lastname` attribute in the migrationUsers.xml file. Otherwise, you will not be able to run the **`XMLaccess`** command. The value for the `lastname` attribute can be any value. The definition and use of the migrationUsers.xml file requires WebSphere Portal administration skills.

14.4.6 Migrating the Administrators group

In previous versions of Lotus Workplace Web Content Management, the Administrators group was granted full access to the Lotus Workplace Web Content Management user interface, including every item and the application backend files. Any user who was a member of this group was granted super user access.

When you migrate users and groups, you have the option to migrate the previous Lotus Workplace Web Content Management Administrators group. If you migrate the Administrators group, the migration program does not create a new Administrators group in WebSphere Member Manager because the wcmadmins group already exists. Instead, it copies the members of the Administrators group into the wcmadmins group.

To migrate the Administrators group, select **Import Administrators Group** in the WebSphere Portal migration page.

14.4.7 Migrating LDAP users and groups

If you used an LDAP server to manage users and groups in Lotus Workplace Web Content Management 2.0, you must still run user migration. However, you must perform one of the following steps prior to migrating users or groups.

- Configure WebSphere Member Manager to use your existing LDAP server.
- Copy your existing LDAP users or groups into the LDAP server being used by WebSphere Member Manager.

Important: You cannot migrate custom LDAP attributes that were used to map Lotus Workplace Web Content Management profiling values to LDAP users and groups.

14.4.8 UI Access Group

The *UI Access Group* is a predefined group in the previous version of Lotus Workplace Web Content Management. In earlier versions the group was called *AptrixUIAccessGroup*. Members of this group gain access to the user interface of earlier versions. This group no longer exists in IBM Workplace Web Content Management 2.5 and 5.1.

When you migrate users, users who were members of the UI Access Group or AptrixUIAccessGroup become members of the virtual Portal group *all authenticated portal users*. This group has access to the Web Content Management label and the pages on which the authoring and rendering portlets are located.

When you migrate groups, make sure the UI Access Group or AptrixUIAccessGroup is not a member of a group.

14.4.9 ALL USERS group

The *ALL USERS* group is predefined group in previous versions of Lotus Workplace Web Content Management. This group no longer exists in IBM Workplace Web Content Management 2.5 and 5.1.

When you migrate users, users who were members of the ALL USERS group become members of the virtual Portal group *all users*. Members of the *all users* group are granted access to items that do not require restricted access.

14.4.10 Anonymous user

The *Anonymous* user is a predefined user in previous versions of Lotus Workplace Web Content Management. This group no longer exists in IBM Workplace Web Content Management 2.5 and 5.1.

When you migrate users, the Anonymous user maps directly to the virtual Portal anonymous user. Items that granted access to the Lotus Workplace Web Content Management Anonymous user now grant access to the Portal anonymous user.

14.4.11 Member Fixer utility

The *Member Fixer* utility checks the security settings in all IBM Workplace Web Content Management items. When run in fix mode, it removes references to users and groups who do not exist in WebSphere Member Manager and updates user and group names that have been changed in WebSphere Member Manager.

If nonexistent users and groups are referenced in IBM Workplace Web Content Management items, performance might be slowed while they are checked. Performance might be improved if nonexistent users and groups are removed.

Attention: Running the Member Fixer utility in fix mode can cause serious security problems. For example, if a user or group was not migrated correctly or if IBM Workplace Web Content Management is not properly connected to WebSphere Member Manager, valid users and groups might be removed from the security sections in IBM Workplace Web Content Management items.

View a report of users and groups using the following URL:

```
http://[HOST]:[PORT]/wps/wcm/connect/?MOD=MemberFixer
```

Remove references to nonexistent users and groups, and update changes to users and groups in IBM Workplace Web Content Management items using the following URL:

```
http://[HOST]:[PORT]/wps/wcm/connect/?MOD=MemberFixer&fix=true
```

Archived

14.5 Verifying and troubleshooting user and group migration

This section guides you in verifying the success of user and group migration and in identifying the cause of errors.

14.5.1 Verifying user migration

To verify that the user migration is successful, take a random sample of users and groups and do the following:

- Open WebSphere Portal administration to verify the creation of users and groups.
- Open the IBM Workplace Web Content Management authoring portlet, and open a content item. Verify that expected categories and keywords appear in the Profile section.
- Log on to the authoring portlet as different users to verify that security and access controls are working.
- Log on to a rendered site as different site visitors to verify access to rendered content.
- Run the User Reference tool in *non-update* mode to make sure that IBM Workplace Web Content Management and WebSphere Member Manager data matches, as in the following example:

  ```
  http://[HOST]:[PORT]/wps/wcm/connect?MOD=AJPEReferenceChecker
  ```

14.5.2 Using the log files

Table 14-5 lists the log files and the types of errors that they report.

Table 14-5 Log files containing user migration errors

Log file	Contents
SystemErr.log	Contains exceptions
Wps_<date>.log	Contains JSP compilation exceptions
SystemOut.log	Contains startup errors

The log files are located in the *WPS_HOME*\log folder.

Archived

14.5.3 Examining common errors and resolutions

Table 14-6 lists the errors that might appear in the browser and possible ways to resolve the problem.

Table 14-6 Common errors and resolutions

Error	Problem and resolution
A compilation error appears when you access the migration interface.	Problem: The ilwwcm-migration.jar was not added to the `WcmLib` classpath field. Resolution: Add the ilwwcm-migration.jar to the WcmLib classpath field.
The migration process fails.	Problem: User and group names contain illegal characters (for example, [,] and *). Resolution: Change the user and group names in the previous version of Lotus Workplace Web Content Management to contain only acceptable characters.
A file not found exception error appears when you start the migration process in the migration interface.	Problem: The users in the *ILWWCM_HOME*\connect folder are not found. Resolution: ► Make sure that you specified the correct *ILWWCM_HOME*\connect folder in the Input Connect User Path field in the migration interface. ► Make sure that all users are in the *ILWWCM_HOME*\connect folder. ► Run the UserChecker utility with `create=true` in the previous version of Lotus Workplace Web Content Management and then copy the users to IBM Workplace Web Content Management 2.5 or 5.1.
The sn attribute for each user does not contain a value.	Problem: Users and groups in earlier versions of Lotus Workplace Web Content Management do not provide a `lastname` attribute, which results in an empty `sn` attribute. This occurs only when using direct migration. Resolution: Migrate users and groups using a user migration mapping file and run the **`XMLaccess`** command. You must type a value for the `lastname` attribute in the migrationUsers.xml file. Otherwise, you will not be able to run the **`XMLaccess`** command. The value for the `lastname` attribute can be any value.

Archived

Error	Problem and resolution
You cannot see the data in the authoring portlet.	Problem: You have not run the user migration process. Resolution: Run the user migration process using the migration interface.
You cannot see data in the authoring portlet, but the header indicates that there are several pages of data.	Problem: The data has not been configured to use the WebSphere Member Manager users and groups. Resolution: Repeat the user migration process using the migration interface.

You cannot access content even though you have the proper security access levels.	Problem: The names provided in the user migration mapping file do not match the correct LDAP or WebSphere Member Manager users. This occurs only when migrating users using the user migration mapping file.
	
	Resolution: Make sure that you are using the fully qualified distinguished name in the migrationUsers.properties file. If WebSphere Portal security is disabled, the distinguished name is one of the following:
	`uid=<user>,o=Default Organization`
	`cn=<group>,o=Default Organization`
	Update the file and select Use User Migration Mapping File Option in the migration interface.

Users do not have the proper categories or keywords.	Problem: The categories and keywords have not been added. This occurs only when migrating users using the user migration mapping file. Resolution: Execute **`XMLaccess`** using the *ILWWCM_HOME*\migration\migrationUsers.xml input file.

Archived

Error	Problem and resolution

When editing and saving content, the security mappings disappear.	Problem: The names provided in the user migration mapping file do not match the correct LDAP or WebSphere Member Manager users. This occurs only when migrating users using the user migration mapping file.
	
	Resolution: Make sure that you are using the fully qualified distinguished name in migrationUsers.properties. If Portal security is disabled, the distinguished name is
	`uid=<user>,o=Default Organization`
	`cn=<group>,o=Default Organization`
	Update the file and select Use User Migration Mapping File Option in the migration interface.

You cannot see the data in the authoring portlet, but the migration was successful.	Problem: The indexes have not been updated with the newly configured data. Resolution: Stop the WebSphere Portal server, delete the indexes, and start the WebSphere Portal server to rebuild the indexes.

14.5.4 Files needed by the IBM Support Desk

If you encounter problems that you cannot solve and if you have a support contract with the IBM Support Desk, send the following files or folders to the support desk:

- Data folder located in *ILWWCM_HOME*\ilwwcm
- Users folder located in *ILWWCM_HOME*\connect
- Wps_<*date*>Log located in *WPS_HOME*\log
- SystemOut.log located in *WPS_HOME*\log
- SystemErr.log located in *WPS_HOME*\log
- Connect.log with trace level 3 located in *ILWWCM_HOME*\connect\log

14.6 Creating the unrestricted resources

Resources, which comprise Web page elements such images, are part of the data transfer from one repository to another. Resources that require restricted access appear on a Web page only after a site visitor logs on. Resources that do not require restricted access appear when anonymous and logged on site visitors view the Web page. When you transfer the resources to IBM Workplace Web Content Management 2.5 or 5.1, unrestricted resources require additional

step before they will appear on a Web page, because they are stored and accessed differently than restricted resources.

14.6.1 Where resources are stored

IBM Workplace Web Content Management stores resources that do not require restricted access in the following two locations:

- In the specified resources repository, defined by the parameter `resource.persistence` in the aptrixjpe.properties file.
- In the *WEB_APP_HOME*\resources folder.

When file system data is transferred to a Cloudscape database, the *WEB_APP_HOME*\resources folder is not created automatically. You need to create it using the Resource Checker utility.

IBM Workplace Web Content Management stores resources that require restricted access in the specified resources repository only.

14.6.2 Accessing unrestricted resources

When a Web page opens, it accesses unrestricted resources from the *WEB_APP_HOME*\resources folder. The Web page accesses restricted resources from the repository, but it cannot access unrestricted resources from the resources repository even though they are also stored there. When you test if the data transfer and user migration was successful, you will preview a Web page. You are prompted to log on, which provides access to the restricted resources. However, the unrestricted resources do not appear on the Web page.

Figure 14-2 on page 586 shows a previewed Web page with an image placeholder in place of the image, because the image does not require restricted access and the *WEB_APP_HOME*\resources folder does not exist after the data transfer.

Archived

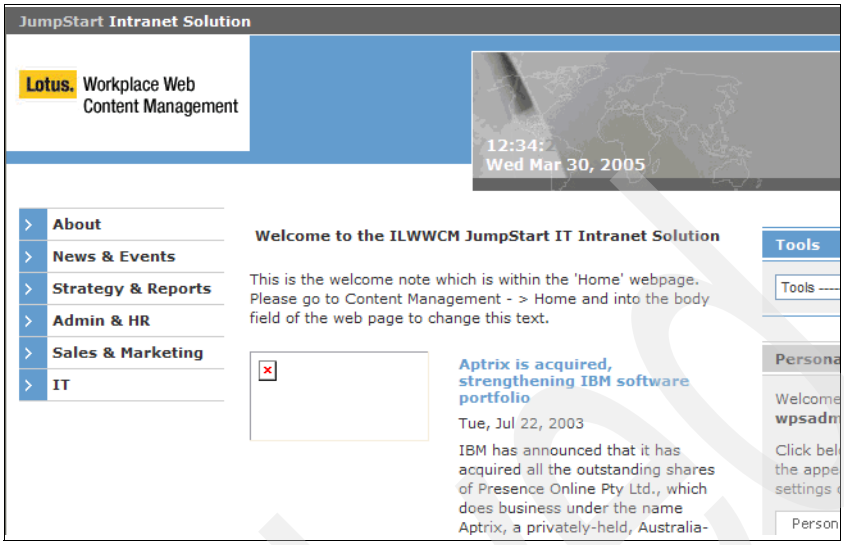


Figure 14-2 Image place holder in place of an image

14.6.3 Resource Checker utility

The Resource Checker, when run in fix mode, creates the *WEB_APP_HOME*\resources folder and copies the unrestricted resources in the repository and recreates them in this folder. Recreate the resources using the following URL:

```
http://[HOST]:[PORT]/wps/wcm/connect?MOD=AJPEResourceChecker&fix=true
```

Archived

14.7 Migrating data from a previous version

This section explains the process for migrating data from a previous version of IBM Workplace Web Content Management.

14.7.1 Transferring file system data to a Cloudscape database

To transfer file system data to a Cloudscape database, do the following:

1. Copy the data folder from the *ILWWCM_HOME*\ilwwcm folder of the previous version of Lotus Workplace Web Content Management to the *ILWWCM_HOME*\ilwwcm folder of IBM Workplace Web Content Management 2.5 or 5.1, as illustrated in Figure 14-3.

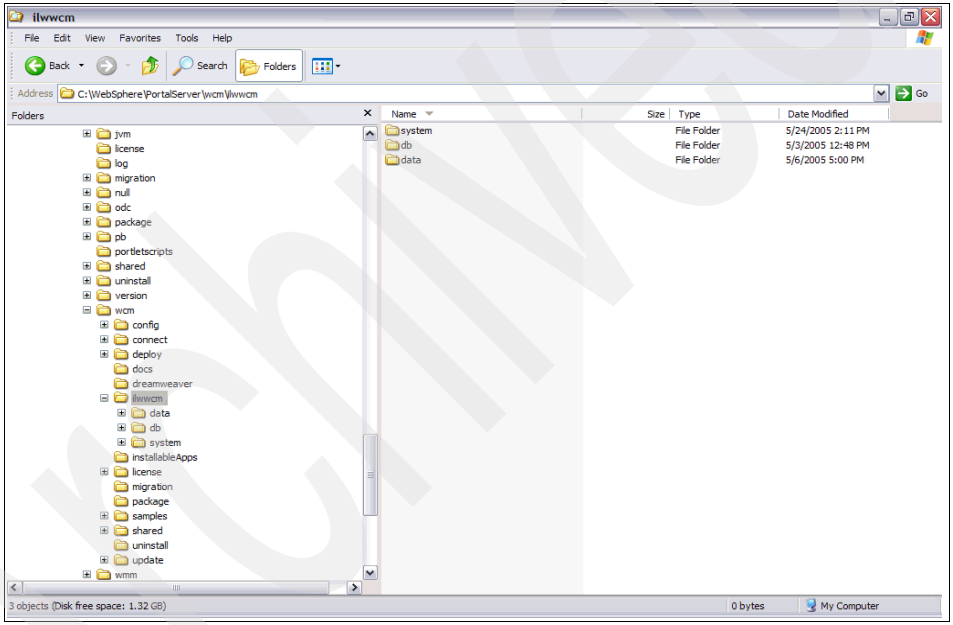


Figure 14-3 Copy the previous version's data folder to ILWWCM_HOME\ilwwcm

Archived

2. Open the wpconfig.properties file in a text editor, as shown in Figure 14-4.

> **Note:** The wpconfig.properties file is located in *WPS_HOME*\config.

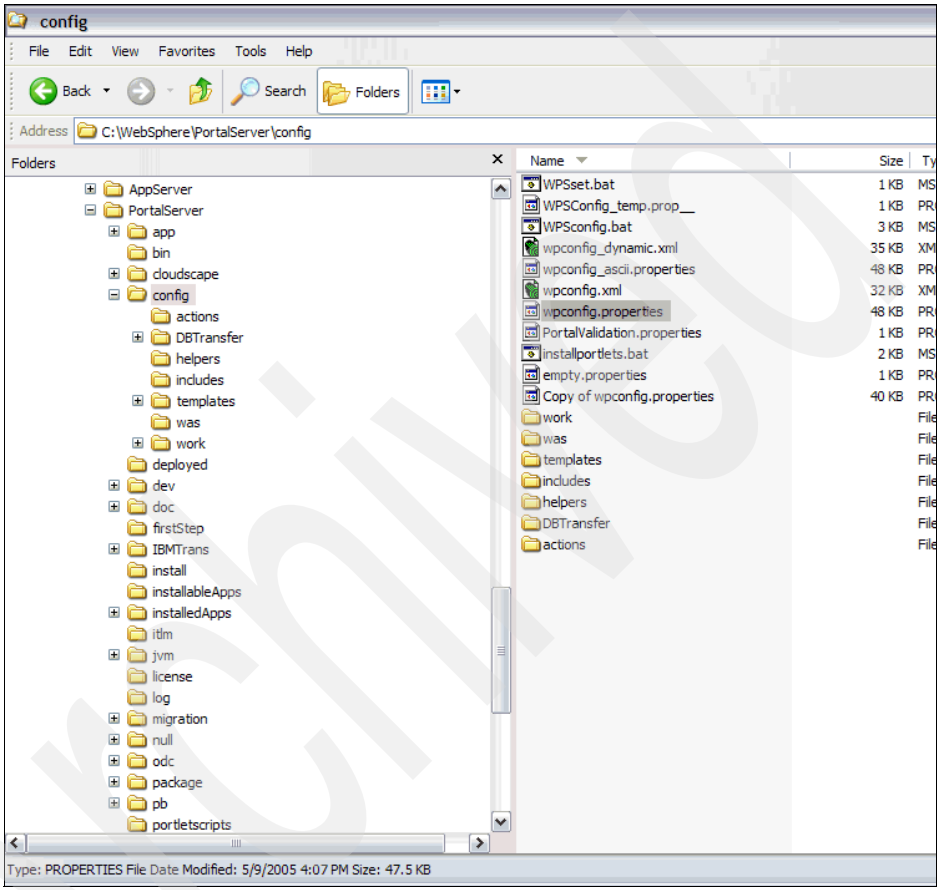


Figure 14-4 Navigate to wpconfig.properties file

3. Find the following line of code:

```
WcmFileSystemDataLocation=
```

4. Enter the path to the file system data files that you copied in the previous procedure (Figure 14-5). The following line of code is an example of how the code might appear:

```
WcmFileSystemDataLocation=C:/WebSphere/PortalServer/wcm/ilwwcm/data
```

```
ReplaceWithYourPassword

pIdShort: The WCM admin group ID
dShort=wcmadmins

e WCM database URL
tabase element of this value should match the value of WcmDbName
S, the database element of this value should match the location
f the database to be used when establishing connections with
2 datasource used by WCM
udscape for Windows, the path in the WcmDbUrl must not include spaces.
he short path for Windows.
  { jdbc:db2j:C:/path/to/WCMDB }
  { jdbc:db2:WCMDB }
  { jdbc:db2:*LOCAL/QWCMDB }
(remote):  { jdbc:as400:<hostName>/QWCMDB }
  { jdbc:db2:<location> }
ote db):      { jdbc:db2://<server address>:<port>/<location> }
  { jdbc:oracle:thin:@<YourDatabaseServer>:1521:WCMDB }
  { jdbc:microsoft:sqlserver://hostname:1433;User=wcmuser;Password=password;SelectMethod=curs
db2j:C:/WEBSPH~1/PORTAL~1/wcm/ilwwcm/db/WCMDB

DataLocation: The location of the ILWWCM 2.5 filesystem data directory to transfer
 only used when migrating from 2.5 filesystem.
ataLocation=C:/WebSphere/PortalServer/wcm/ilwwcm/data

 The name of class SqlProcessor will use to import SQL files, also known as "JDBC provider"
  { com.ibm.db2j.jdbc.DB2jDriver }
  { COM.ibm.db2.jdbc.app.DB2Driver }
  { com.ibm.db2.jdbc.app.DB2Driver }
(remote db):  { com.ibm.as400.access.AS400JDBCDriver }
  { com.ibm.db2.jcc.DB2Driver }
ote db):      {com.ibm.db2.jcc.DB2Driver }
  { oracle.jdbc.driver.OracleDriver }
  { com.microsoft.jdbc.sqlserver.SQLServerDriver }
m.ibm.db2j.jdbc.DB2jDriver

s: The name of class SqlProcessor will use to import SQL files via data source
  { com.ibm.db2.jcc.DB2ConnectionPoolDataSource }
ote db):      { com.ibm.db2.jcc.DB2ConnectionPoolDataSource }
  {com.ibm.db2.jdbc.app.UDBXADataSource}
com.ibm.db2j.jcc.DB2ConnecitonPooldataSource
```

Figure 14-5 The path to the file system data files

5. Make sure that all the IBM Workplace Web Content Management parameters have values and that the values are correct, particularly the password for the `WcmDbPassword` parameter.

> **Note:** The four IBM Workplace Web Content Management parameters for Content Manager do not require values.

6. Save and close the wpconfig.properties file.
7. Stop the WebSphere_Portal application server.
8. Start the WebSphere_Portal application server.
9. Open a command prompt.
10. Change the directory to the *WPS_HOME*\config directory.

Archived

> **Note:** Make sure you change to the correct folder on your system. For example, Portal server might be installed on the D drive.

11. At the *WPS_HOME*\config prompt, enter the following line of code for the Windows platform:

    ```
    wpsconfig.bat transfer-wcm-filesystem-to-cloudscape
    ```

    The file system data is transferred to the Cloudscape database and a `Build Successful` message appears in the command prompt window.

    > **Note:** For the UNIX platform, enter the following:
    >
    > ```
    > wpsconfig.bat/.sh transfer-wcm-filesystem-to-cloudscape
    > ```

12. Close the command prompt window.
13. Open the wcm_migration.log file from the *WPS_HOME*\log folder in a text editor.

    > **Note:** The log file should contain the text `Utility Completed` to confirm that the data transfer was successful.

14. Close the wcm_migration.log file.

14.8 Enabling user migration

This section explains how to install the migration .jar to enable user migration.

> **Important:** In IBM Workplace Web Content Management 5.1, you might need to add the ilwwcm-migration.jar file to your installation before performing this procedure. Read the product release notes for the latest information.

To install the migration .jar file, do the following:

1. Start server1 of the WebSphere Application Server, if it is not already started.
2. Open the WebSphere Application Server Administrative Console.

   > **Note:** Use port 9090 when accessing the WebSphere Application Server Administrative Console. Otherwise, you will not have access to the Shared Libraries.

Archived

3. Click the Environment navigator.
4. Click **Shared Libraries**.
5. Click **WCMLib**.
6. In the Classpath field, enter the full path to the ilwwcm-migration.jar. In a standard installation the code appears as follows:

   ```
   ${WPS_HOME}/wcm/migration/ilwwcm-migration.jar
   ```

7. Click **Apply**.
8. Click **Save**.
9. Click **Save** to save the master configuration.
10. Stop the WebSphere_Portal application server.
11. Start the WebSphere_Portal application server.

14.9 Migrating users directly to WebSphere Member Manager

To migrate users directly to WebSphere Member Manager, do the following:

1. Copy existing users from the previous version of Lotus Workplace Web Content Management to IBM Workplace Web Content Management 2.5 or 5.1 by copying the users folder from the *ILWWCM_HOME*\connect folder of the previous version of Lotus Workplace Web Content Management to the *ILWWCM_HOME*\connect folder of IBM Workplace Web Content Management 2.5 or 5.1.

   Note: The connect folder contains the users folder, which is needed for data migration.

2. Migrate existing users by first starting the WebSphere Portal server, if it is not already started.
3. In the browser Address field, enter the following URL to access the IBM WebSphere Portal migration user interface:

   ```
   http://[HOST]:[PORT]/wps/wcm/jsp/migration/migration.jsp
   ```

4. In the Input WebSphere Portal Administrator Login Name field, enter the Portal administrator name.
5. In the Input Connect User Path field, enter the path to the copied *ILWWCM_HOME*\connect\users folder.

Note: You copied these users from the previous version of IBM Workplace Web Content Management.

6. Select **Import Administrators Group** if you need to migrate the Administrators group.

   Note: The Administrators group is mapped to the Administrators group that is defined in the connect.cfg file. The default value for this group is wcmadmins.

7. Make sure that **Update WMM?** is selected.

   Note: The Update WMM? option is selected by default.

8. Click **Migrate**.

   The users and groups are created in WebSphere Member Manager and the transferred IBM Workplace Web Content Management data points to the new users and groups in WebSphere Member Manager.

9. Stop the WebSphere Portal server.
10. Delete the system folder located in the *ILWWCM_HOME*\ilwwcm folder.

    Note: The system folder contains the indexes.

11. Start the WebSphere Portal server.

    The indexes are rebuilt.

12. To verify that the migration was successful, open the IBM Workplace interface and log on.
13. Select the Web Content Management tab.
14. Expand the Content Library view.
15. Click **Content by Title**.
16. Click a content document.

    The content document opens in the authoring portlet.

17. Expand the Security section of the content document.

    Make sure that the user and group names appear in the Security section. If the names appear, the migration was successful.

Archived

14.10 Migrating users to WebSphere Member Manager using a user migration mapping file

This section explains how to:

- Generate a user migration mapping file.
- Review and edit, if required, the user migration mapping file.
- Process the user migration mapping file.
- Verify that user migration was successful.

14.10.1 Generating a user migration mapping file

> **Tip:** Follow these steps to migrate Lotus Workplace Web Content Management users and groups using a user migration mapping file if the match is not exact between existing Lotus Workplace Web Content Management users and groups and existing WebSphere Member Manager users and groups.

To generate a user migration mapping file, do the following:

1. In the browser type the following URL in the Address field to access the IBM WebSphere Portal migration user interface:

   `http://[HOST]:[PORT]/wps/wcm/jsp/migration/migration.jsp`

2. In the Input WebSphere Portal Administrator Login Name field, enter the Portal administrator name.
3. In the Input Connect User Path field, enter the path to the copied *ILWWCM_HOME*\connect\users folder.

   > **Note:** These are the users that you copied from the previous version of IBM Workplace Web Content Management.

4. Select **Import Administrators Group** to migrate the Administrators group.

   Members of the Administrators group are added to the wcmadmins group.

5. Click **Generate User Migration Mapping File**.

   This option allows you modify the user and group names, as shown in Figure 14-6 on page 594.

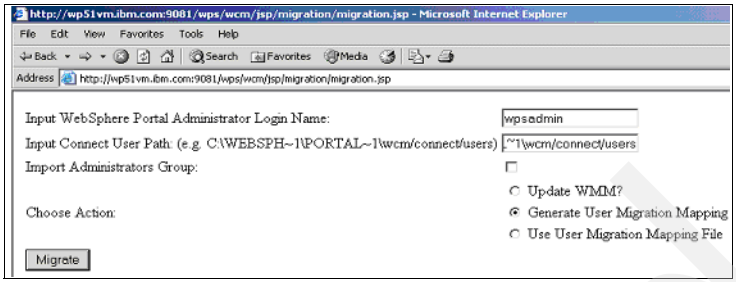


Figure 14-6 Option to generate a User Migration Mapping File

6. Click **Migrate**.

   The user migration mapping file is generated. The file is called migrationUsers.properties and is located in the *WPS_HOME*\wcm\migration folder.

   Note: Do not close the browser window.

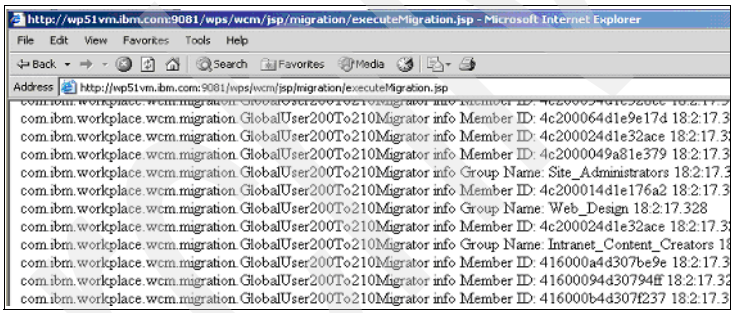


Figure 14-7 Status of the migration

14.10.2 Reviewing and editing the user migration mapping file

To review and edit the user migration mapping file, do the following:

1. From the *WPS_HOME*\wcm\migration folder, open the migrationUsers.properties file in a text editor.
2. If required, modify the user names in the migrationUsers.properties file as follows:
   – The original Lotus Workplace Web Content Management users and groups are listed on the left, and the WebSphere Member Manager users and groups are listed on the right.

– Change only the proposed WebSphere Member Manager names.

– Change the property values of the WebSphere Member Manager name to use the distinguished name. For example:

```
uid=username,o=default organization
cn=group name,o=default organization
```

– Do not use an ampersand (&) in user or group names because it causes errors.

The following line of code is an example of how to change the group name *All Staff* to *All_WP_Staff*, using the distinguished name:

```
All\ Staff=cn=All_WP_Staff,ou=groups,dc=ibm,dc=com
```

Note: Make sure you modify this file before processing it.

3. Save and close the file.

14.10.3 Processing the user migration mapping file

To process the user migration mapping file, do the following:

1. In the open browser window, click **Back**.
2. Click **Use User Migration Mapping File**.
3. Click **Migrate**.

   The browser window shows that the data has been updated with the new user and group names.
4. Stop the WebSphere Portal application server.
5. Delete the system folder that is located in the *WPS_HOME*\wcm\ilwwcm folder.

   Note: The system folder contains the indexes.

6. Start the WebSphere Portal application server.

   The indexes are rebuilt.

14.10.4 Verifying that the migration was successful

To verify that the migration was successful, do the following:

1. Open the Workplace interface and log on.
2. Click **Web Content Management**.

3. Expand the Content Library view.
4. Click **Content by Title**.
5. Click a content document.

   The content document opens in the authoring portlet.
6. Expand the Security section of the content document.

> **Note:** Make sure that the user and group names appear in the Security section. If the names appear, the migration was successful.

14.10.5 Adding new users or groups to WebSphere Member Manager

> **Important:** The user migration mapping file updates only existing Lotus Workplace Web Content Management users and groups. If existing Lotus Workplace Web Content Management users and groups do not exist as WebSphere Member Manager users and groups, you must run the **`XMLaccess`** command.
>
> Users and groups in earlier versions of Lotus Workplace Web Content Management do not provide a `lastname` attribute. You must type a value for the `lastname` attribute in the migrationUsers.xml file. Otherwise, you will not be able to run the **`XMLaccess`** command. The value for the `lastname` attribute can be any value.

To add new users or groups to WebSphere Member Manager, do the following:

1. Open a command prompt.
2. Go to /PortalServer/bin.
3. Run the following command:

```
xmlaccess.bat -in /path/PortalServer/wcm/migration/migrationUsers.xml -user
portalAdmin -pwd password -url http://host:port/wps/config
```

> **Note:** Ensure that you replace path, portalAdmin, password, host, and port with the actual setting of your IBM Workplace Web Content Management installation.
>
> This command is only usable in a Windows platform. For the UNIX platform, use the following command:
>
> ```
> xmlaccess.sh -in /path/PortalServer/wcm/migration/migrationUsers.xml
> -user portalAdmin -pwd password -url http://host:port/wps/config
> ```

14.11 Disabling user migration

To uninstall the migration .jar file to disable user migration, do the following:

1. Start server1 of the WebSphere Application Server, if it is not already started.
2. Open the WebSphere Application Server Administrative Console.

> **Note:** Use port 9090 when accessing the WebSphere Application Server Administrative Console. Otherwise, you will not have access to the Shared Libraries.

3. Click the **Environment** navigator.
4. Click **Shared Libraries**.
5. Click **WCMLib**.
6. In the Classpath field, delete the full path to the ilwwcm-migration.jar. In a standard installation, the code appears as follows:

```
${WPS_HOME}/wcm/migration/ilwwcm-migration.jar
```

7. Click **Apply**.
8. Click **Save**.
9. Click **Save** to save the master configuration.
10. Stop the WebSphere Portal application server.
11. Start the WebSphere Portal application server.

14.11.1 Recreating the resources

To recreate the resource folder and the resources, enter the following URL in the browser Address field:

```
http://[HOST]:[PORT]/wps/wcm/connect?MOD=AJPEResourceChecker&fix=true
```

The Resource Checker runs. When completed, the log appears in the browser window.

Archived

Archived

15

Migrating content into IBM Workplace Web Content Management

This chapter describes how to migrate a Web site from another Web content management tool into IBM Workplace Web Content Management 5.1. It includes the following sections:

- 15.1, “A roadmap to migrating Web content” on page 600
- 15.2, “Overview of the Vamosa Content Migrator” on page 609

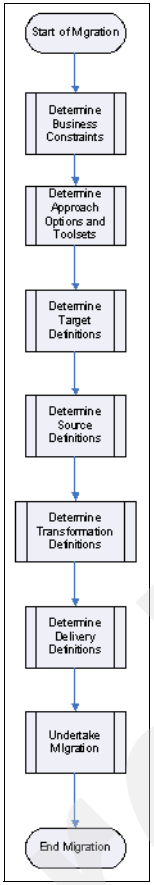
Note: For the purposes of this chapter, we highlight Vamosa’s Content Migrator tool set. For additional information, you can refer to the following:

http://www.vamosa.com/index/what_we_do/technology_and_products/vamosa_content_migrator.htm

Note that Vamosa’s technology represents one of the strategic options for migrating content into IBM Workplace Web Content Management. However, other technical approaches are available. Regardless of which technology that you use to migrate content, the content in 15.1.1, “Determining the business case for the migration” on page 601 through 15.1.6, “Determining quality assurance measures” on page 607 includes key fundamental business issues that you need to address in your migration.

© Copyright IBM Corp. 2006. All rights reserved.

15.1 A roadmap to migrating Web content



Undertaking a migration of Web pages or documents into a content management system (CMS) is often approached as a *last minute* exercise. Normally, assurances from CMS vendors imply that there are no issues that are associated with this part of implementing a successful CMS strategy. However, as with many other aspects of IT implementations, it is the correctness and validity of the data that is entered into a system that determines its success or failure.

Thus, IBM has teamed with Vamosa to recommend the use of Vamosa's unique toolset to migrate all of an organization's existing content into IBM Workplace Web Content Management. As CMS has matured as a sub-set of IT systems, the provision of a successful and proven migration methodology has become more important in determining the suitability of one CMS product over another.

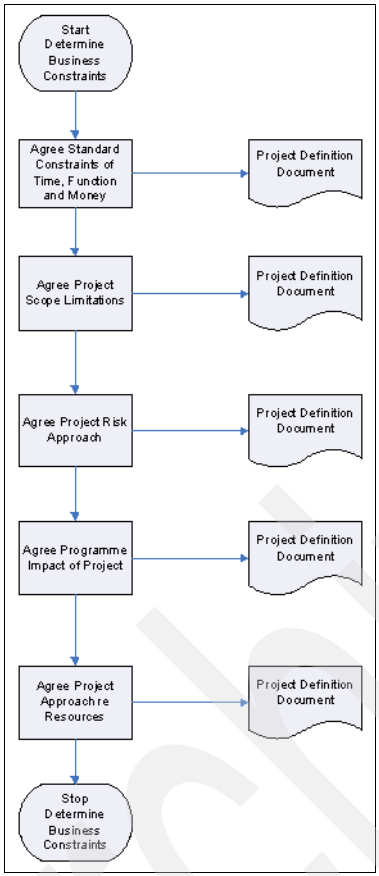
Conceptually, it is not difficult to understand how to migrate existing data, in that you can visualize the end result easily. However, when trying to understand the process steps that are required to move data from the existing environment to the target CMS system, difficulties arise. In these cases, a roadmap that shows the route for the migration and that contains some simple rules can help you achieve a successful migration. This section demonstrates a roadmap and provides the rules that are required.

The roadmap requires a number of distinct steps. If you follow these basic steps, the migration will be successful.

Each step in this roadmap has an associated flowchart that outlines the processes that is required within that step. The remainder of this section describes each step.

Archived

15.1.1 Determining the business case for the migration



As with all projects, you should begin your migration project by outlining the business case for the migration. This business case is constrained by time, cost, and function. These constraints should be embedded within the philosophy of the migration, and they should be reflected within the project operation.

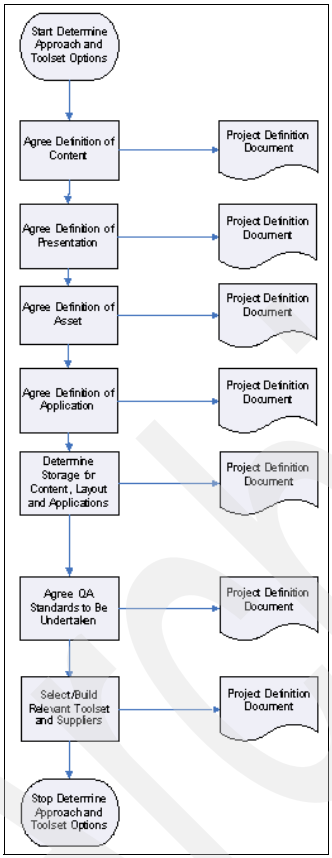
The measure of success for the migration requires that you set the business case clearly at this stage of the process. The migration requirement does not impinge into the area of business change. Instead, the business change project should actively set the boundaries for the migration project.

Vamosa Implementation Architecture (VIA) is the best practice content migration method. It is designed to leverage the maximum automation benefit from Vamosa Content Migrator. VIA defines the means to extract, document, and implement the valid business constraints for the migration project.

Archived

15.1.2 Determining the migration approach and toolset

Based upon the constraints placed upon the project it is necessary to identify and ratify the migration approach and required toolset necessary to satisfy the objectives. If the volume of data that you need to migrate is significant, then an automated approach is the best solution. However, if the volume is smaller — less than 5000 pages — then a manual approach is sufficient.



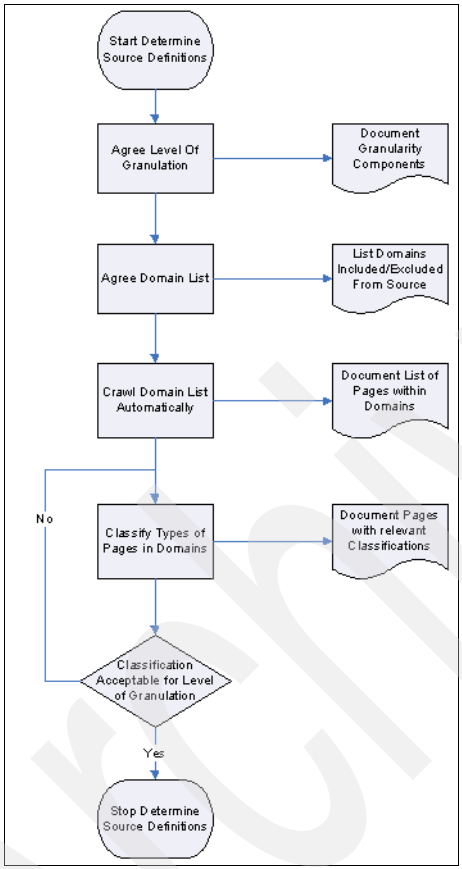
Migrations throw up a number of options that can be exploited. This is even more prevalent when you use an automated approach. These options fall into four distinct types, with each type being a super-set of the type before it, as follows:

- *As-is migration*. In this case, the source pages and assets are migrated with no change to their nature and no additional information is added to their definition.
- *Enhanced migration*. In this case, the source pages have additional metadata that is created with regards to their properties and also to the requirements of IBM Workplace Web Content Management.
- *Standardized migration*. In this case, the input pages are broken down into smaller levels of granularity, and these smaller levels are migrated and re-assembled in IBM Workplace Web Content Management. This case is often referred to as *applying standard templates*.
- *Restructured migration*. In this case, as the pages are migrated, then the navigation elements are updated to reflect a new information architecture.

At this stage, it is important to ensure that the original constraints are *built into* these decisions. All migrations as they progress tend to attract a growing requirements list. This scope creep occurs if the original constraints — and the business requirements upon which they are founded — are ignored. It is important to tie back the choices for which type of migration is undertaken based upon those original constraints and the success criteria that is designed for them.

15.1.3 Determining source classifications for the migration

Often within migrations, the quality of the input data is ignored until later on in the project. A successful migration strategy determines the source classifications to be assured that the current data can fit into the target formats.



The following are the aspects that you need to define for the source information:

- Domains
- Information architecture
- List of pages and assets
- Number of page classifications
- Available granularity by classification

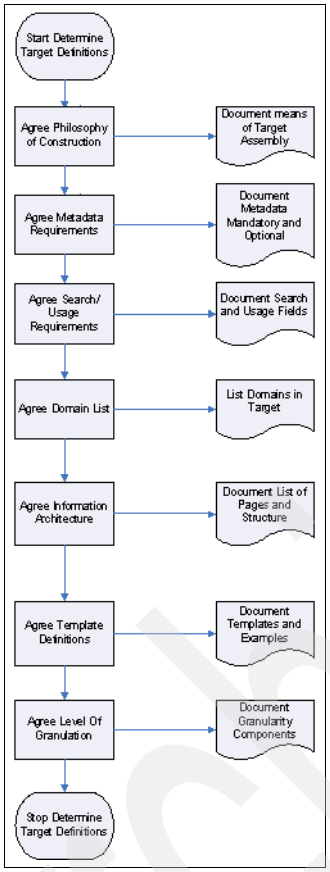
By defining these elements, you can determine the potential complexity of the migration. Separation of pages and assets is crucial at this stage. All images and documents that are linked from pages are categorized as assets and need to be recorded as such. Classification of the input pages and assets allows the migration strategy to take account of all input parameters, which is vital to achieving the necessary granularity.

Based upon our experience when using Vamosa Content Migrator, in any single Web site migration, there are typically a limited number of source classifications because the actual pages conform to a subtle presentation format that is acceptable to the users who interact with them. If the formats are too numerous or are not related, then the current system is not usable and an automated content migration is unfeasible.

We have found that, depending on the size of the source volume, there is a maximum of 20 to 30 classification types. You can investigate these classification types manually. For example, you can investigate a sample page from each type to best determine the granularity that can be achieved during the retrieval process.

Archived

15.1.4 Determining the target definition for the migration



As part of any migration, the definition of the target system determines the transformation requirements. At this stage, organizations can fall easily into designing a complicated solution that cannot be implemented. Consequently, there is a close relationship between the emergent information regarding the source classification types and the potential for populating the designed target system successfully.

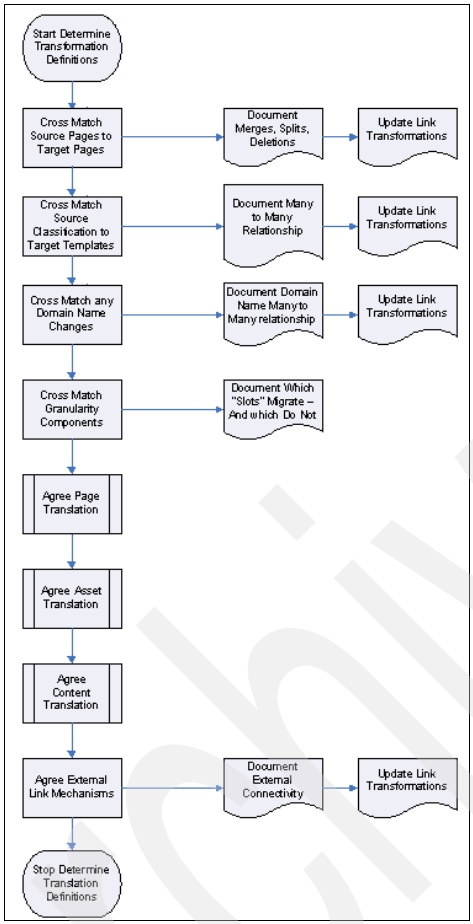
You need to match the definitions in the source environment with those in the target environment. However, you can assume safely that the requirements of the target system are greater than those enabled currently in the source system. These requirements include the following:

- Domain hierarchy
- Information architecture
- Template definitions
- Granularity specifications
- Metadata requirements

You should remember that as part of this process you need to incorporate the requirements of the IBM Workplace Web Content Management implementation, including aspects such as naming conventions, ownership, security, expiration dates, classifications, or taxonomies and so forth.

Archived

15.1.5 Determining the transformation requirements for the migration



Now that you have defined both the source and target systems, you can determine the transformation rules for the migration. These rules need to address all aspects of the target system as the primary recipient of the existing data. Subsequently, when determining the rules, you need to *work backwards* from the target system to the source system. The overall approach is to make the source data *fit* the target data and not to change the target to suit the source.

The definition and the design of the rules, therefore, are driven from the target system requirements. Where the source system cannot supply relevant data to populate the target system, then you need to apply effective auto-generation logic within the rule definitions. In addition, where the target data cannot be populated, either due to inappropriate source data or auto-generation that cannot be completed, then you need to define an exception handling format.

For example, if the target system requires a Title field to be less than 50 characters long but the source system has a title that is 65 characters long, you should document this exception for action. The exception could state that the source is truncated to 50 characters or that the data element is flagged so that it is not migrated until the correct rule handling is defined. The transformation rules can be broken down into the following key areas:

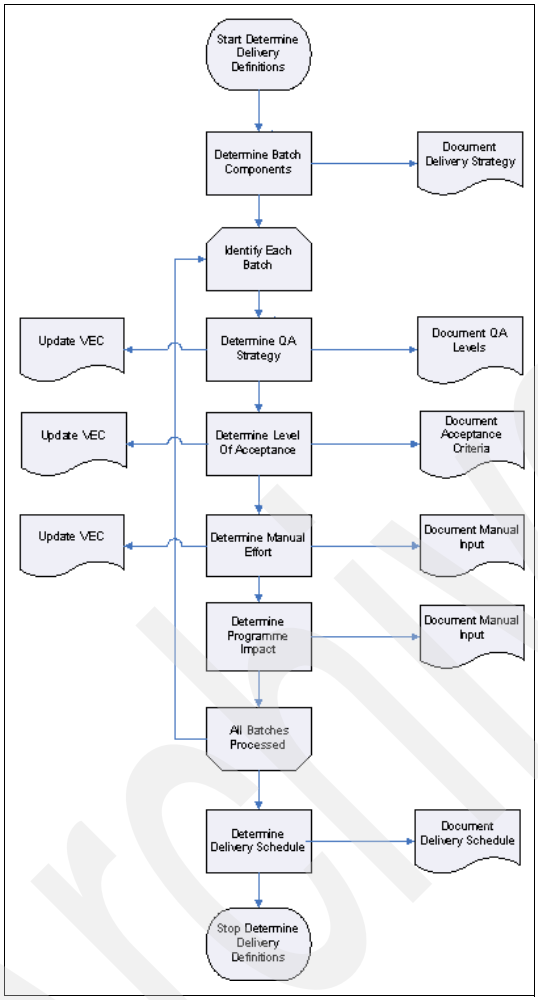
- Domain name
- Information architecture
- Classification match to template
- Granularity cross-matching
- Page changes
- Asset requirements
- Content enhancement

You should design and test these rules prior to actual coding. Using desk-checking of the actual pages that are identified as part of the classification is invaluable in ensuring that the rules work when they are actioned.

You should also associated a verification, exception, and correction document (a VEC document) to each rule. This document determines the test of verification — what exception is identified — and the corrective action to take. In the previous example, the verification checks the title field to see if it conforms to requirements. The exception is that the length is greater than 50 characters. The correction is to truncate the title to the first 50 characters and to dispose of the rest. This action is then incorporated into the conversion rule action.

Archived

15.1.6 Determining quality assurance measures



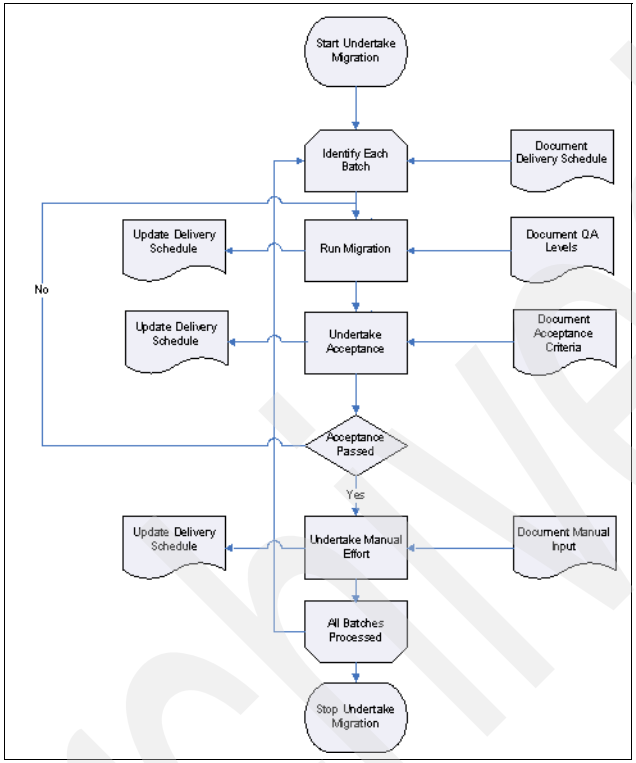
All migrations have a requirement that measures them against success criteria. Where volumes are small, this measurement is not an onerous task, but as volumes increase, then the need for a number of smaller batches arises. Undertaking a migration of one million pages can indeed be achieved in one large scale batch operation. However, the quality assurance function can prove to be difficult when it comes to checking all of these pages.

When you determined the approach and toolset for the migration, you also identified the quality assurance requirements. Now, it is necessary to design an approach that allows you to undertake these quality assurance aspects. Typically, the design involves breaking down the migration into smaller, more manageable sub-projects that are migrated in either a parallel or a sequential manner. If you take this approach, you need to ensure that the relationship between link items is maintained across the sub-projects.

Vamosa Content Migrator is unique in the fact that the toolset supports this dynamic link handling and ensures that the information architecture is maintained.

15.1.7 Undertaking the migration

At this point, the actual undertaking of the migration should be a relatively straightforward task.



The definitions have set out a framework of operation that reflects both source information and the required target system. By defining both the requirements that exist currently and the requirements that are to be, the transformation rules and the exception handling, and the quality assurance that is necessary for the migration, all the technical elements are in place to proceed with the migration.

If during the initial migrations additional facts come to light (that is, any unforeseen data sources), then you should reflect this information further back in the source definition and the transformation requirements. Matching the final migration to the success criteria is extremely important at this stage. It should not be possible to simply reject migrations without identifying those migrations as failing the success criteria based upon the transformation rules that you defined previously.

Archived

15.2 Overview of the Vamosa Content Migrator

This section provides an overview of the Vamosa Content Migrator software product and illustrates step-by-step the process of migrating data using this product.

Vamosa Content Migrator is a unique software product that performs all of the tasks that are involved in the complex process of extracting content from any source, transforming that content to make it compliant at both the technical level and the business level, and then loading that content into IBM Workplace Web Content Management.

15.2.1 How Vamosa Content Migrator works

Vamosa Content Migrator uses a multi-step process to ensure that all content is processed adequately to suit the requirements of IBM Workplace Web Content Management. When migrating a Web site into IBM Workplace Web Content Management, Vamosa Content Migrator uses four stages to achieve the desired results, as described in the following sections.

Step 1: Identifying the content to be migrated

Vamosa Content Migrator can selectively identify the content which is the candidate for the migration project. The VIA methodology describes the best way to break down the project into sub-components, as illustrated in Figure 15-1 on page 610. Each sub-project can then be defined by the URLs that define the content. Vamosa Content Migrator then crawls across this content to allow the Vamosa Business Content Process Management repository to be defined for this specific sub-project.

Archived

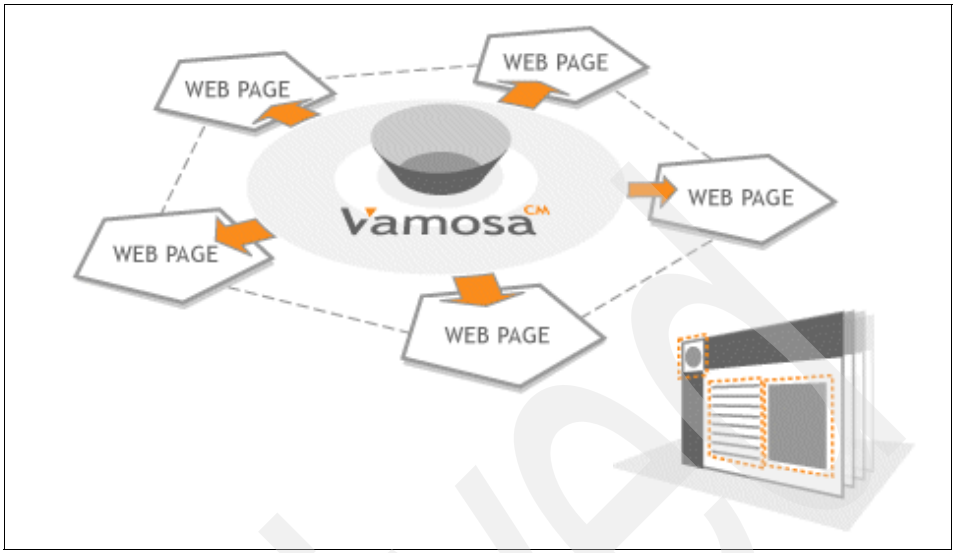


Figure 15-1 Vamosa Content Migrator scans the entire site to break down into parts

This initial phase also uses the Vamosa Content Migrator Source Classifier to categorize the existing content so that the automated migration can take place. The Vamosa Content Migrator Source Classifier generates *source templates* that break down an HTML page into its logical parts, such as navigation, content, header, and footer text. The range of templates generated are then matched to each page during the extract process where the *fragments of HTML code* that make up a Web page are broken down into their constituent parts and held separately within the Vamosa repository. When in the repository, these fragments of HTML code are further processed during the enhancement stage by applying specific business rules to the various fragment types.

As each page passes through the Vamosa Content Migrator Source Classifier, rules help the Vamosa Content Migrator Source Classifier to determine whether or not a particular table cell contains a high degree of links or a high degree of text. The HTML code is then annotated with the classification results, which in turn are read by the extract process to break down the Web page into its constituent parts. This classification is also used to *map* source page types to target application templates.

Step 2: Extracting the content to be migrated

After you have identified the migration candidate content, the next task is to actually take that content and place it in the Vamosa Business Content Process Management repository for further processing, as illustrated in Figure 15-2. The level of granularity that the content is broken down to is a *project parameter*. Vamosa Content Migrator can break content into chunks ranging from pages down to single characters. This flexibility allows you to map the old structure that is identified through the Vamosa Content Migrator Source Classifier into the templates or slots as required by the target system.

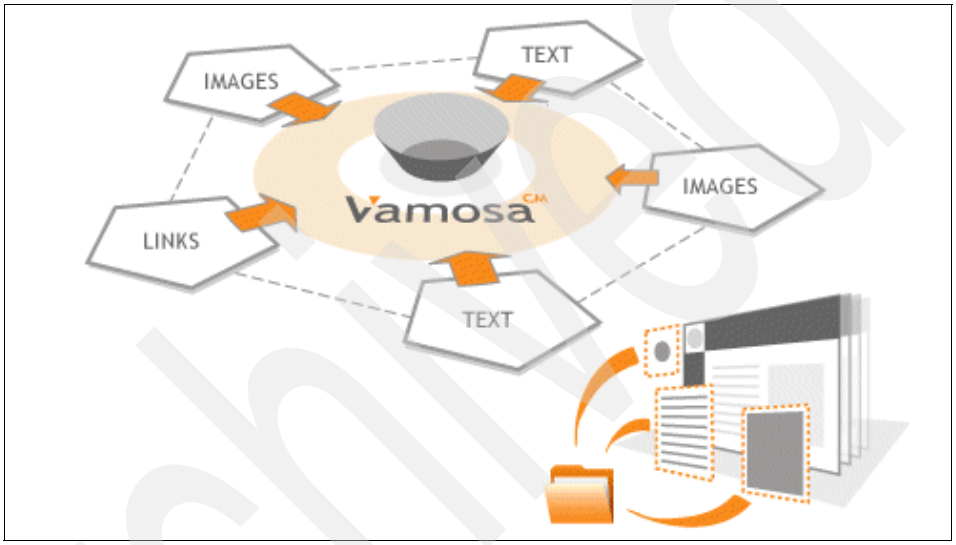


Figure 15-2 Vamosa Content Migrator extracting and storing content

The following sets of data discuss how Vamosa Content Migrator can break content into more manageable portions of data:

- Set 1 captures the relevant structure and context of the existing content. Vamosa captures and stores existing metadata within the repository. Vamosa also creates new metadata for the content that is extracted. This allows Vamosa Content Migrator to process all data based upon the actual value of that data and its source location and relationship to other content. This type of data is referred to as *placeholder data*.
- Set 2 captures all files that are used by the source system. These files include PDFs, presentations, spreadsheets, and word processing files as well as all images, sound, and movie elements. This type of data is referred to as *asset data*.
- Set 3 captures all HTML data from the source system. This type of data is referred to as *content data*.

This data is placed into the Vamosa Business Content Process Management repository for further processing.

> **Note:** Vamosa Content Migrator does *not* alter or destroy the source system. The source system remains completely intact until the target system is ready to move to production.

Step 3: Re-engineering the extracted content to make it fit for its purpose

This step allows for full programmatic access to the stored components of placeholders, assets, and content, as illustrated in Figure 15-3.

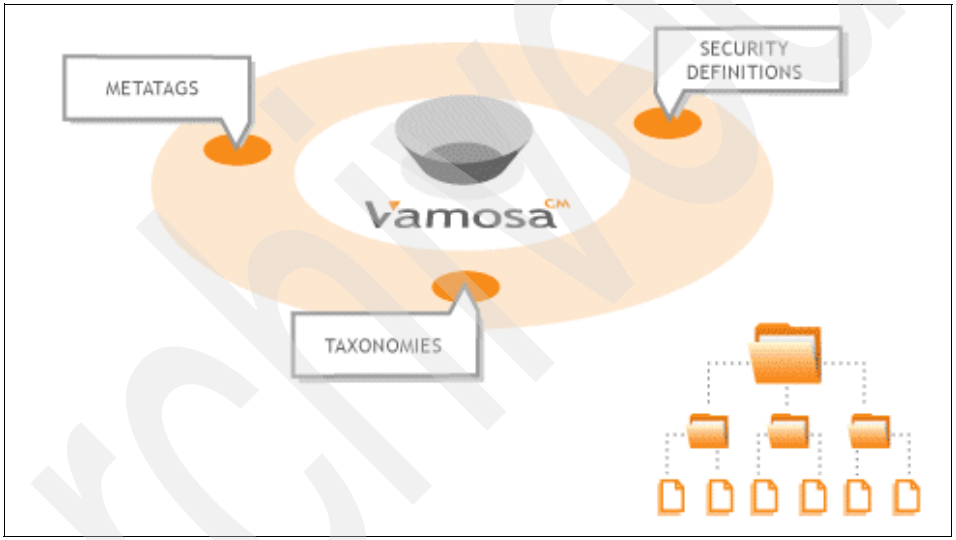


Figure 15-3 Vamosa Content Migrator categorizes Web content into the new structure

To allow customization at this stage, a scripting API is available. Default scripts are provided with Vamosa with the intent of providing a basic enhancement of data to allow loading the content into IBM Workplace Web Content Management. These default scripts are available automatically on the creation of a task library within the Vamosa Content Migrator tool and, generally, can be used as is.

For more complex requirements, however, it is necessary to write (or modify) scripts on a per project basis. You can write scripts using the rule editor that is built into the Vamosa tool or using an external editor and then pasting in the scripts when they are complete.

There are two scripting languages used:

- Jython, a JAVA-enabled Python engine that is available at:

  http://www.jython.org

- XSL, a stylesheet translation language which defines templates to modify the XML content.

Typically, scripts are written to achieve some (or all) of the following:

- HTML pre-processing
- Navigate from initial URL and change fixed links to relative links
- Links process
- Page names
- Page locations
- Page metadata
- Asset names
- Asset locations
- Asset metadata
- Content extract
- Content merging
- Content deletion
- Tag extract
- HTML changes
- Font standardization
- Menu standardization
- Spelling correction
- Text replacement
- De-duplication
- XML post-processing

This list is not exhaustive. The ability for Vamosa to enhance content is limited only by the time that is available within the project to complete these tasks.

Archived

Step 4: Loading content into IBM Workplace Web Content Management

When the content is in the state required by the project, it is *application ready*. The final part of the Vamosa Content Migrator automated process is to load this application ready data into IBM Workplace Web Content Management. Vamosa Content Migrator can achieve this load in a number of ways, but each method achieves the same end results as illustrated in Figure 15-4.

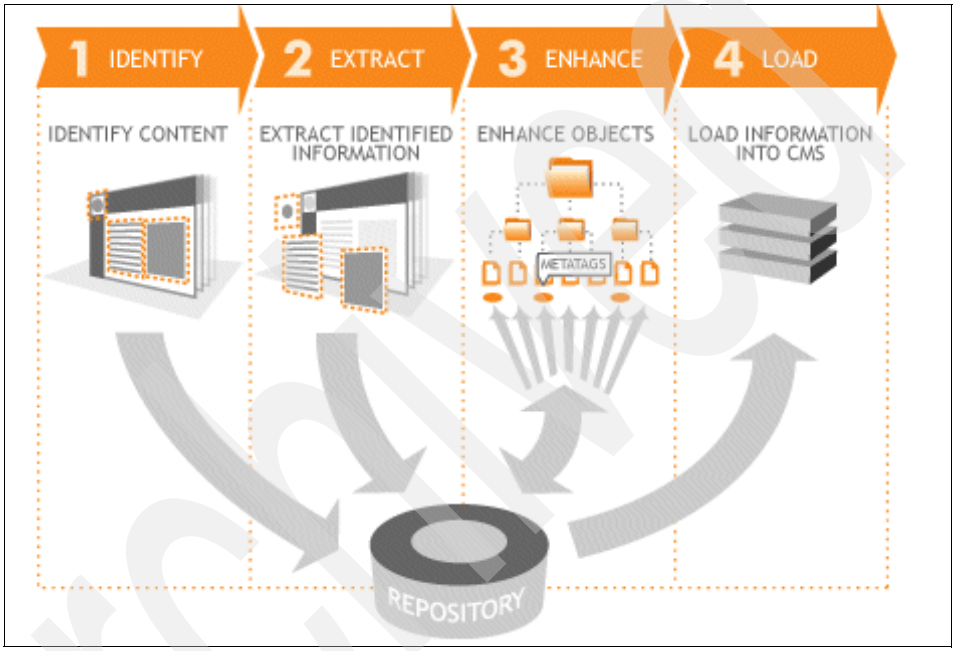


Figure 15-4 Overview of the steps that are involved in loading data

15.2.2 Reasons for using a specialized method

These four stages are the overall process by which data is migrated. As the data is migrated, it needs to be updated and enhanced in order to allow the correct load into IBM Workplace Web Content Management. The overall aspect of VIA is directly focused to this aspect of the migration. Indeed the requirement to specify data enhancement is referred to deliberately as *business* rules. Therefore, VIA has a supporting management infrastructure built in that is designed to control and manage the overall method process steps.

Although this method outlines the steps that are associated with data migration, it does not represent a form of project definition that would work successfully.

Essentially, Figure 15-4 on page 614 demonstrates how data is migrated, but it does not define the process to undertake that migration.

For example, if the target content management system is an IBM Workplace Web Content Management system, it is not permissible to load data that contains basic HTML components, because the HTML components are present in the source system. These HTML components must be *enhanced* to become IBM Workplace Web Content Management components. By following the method, you can answer and build these questions into the migration steps and incorporated them easily into the Vamosa Business Content Process Management engine.

To ensure that specialist knowledge is not required and that service personnel can undertake successful migrations, this method has specific steps .

Figure 15-5 illustrates the VIA method components.

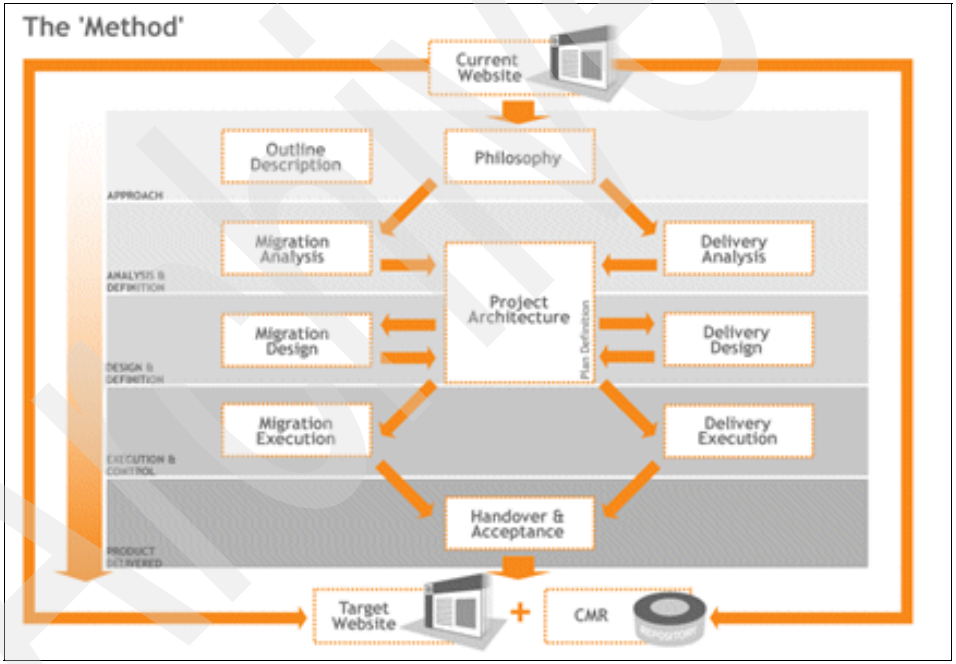


Figure 15-5 VIA method components

15.2.3 The law of diminishing returns

Although the theory of achieving 100% data migration is the goal of any migration, it is usual in most circumstances that data is migrated within the context of iterative migration sets that work towards a 100% success rate.

Typically the *business rules* that are designed and defined exploit a large sub-set of the migrated data. However, after the first set of runs, there is a minor sub-set of data that has not been addressed correctly. Additional business rules are defined, and the four migration steps are run again. This process is repeated until a high enough success rate has been achieved.

This process, however, succumbs to the *law of diminishing returns*. Each iteration achieves a smaller return on the additional successful data that is migrated. As part of the migration process, it is important to define at the initiation of the project what the success levels for the project should be, bearing in mind the law of diminishing return.

15.2.4 Quality assurance and project management

VIA allows for the effective quality control and project management of a content migration project. VIA identifies the processes that are required to generate the necessary checkpoints when running the projects to ensure that all data is processed and accounted for. This identification obviates the requirement to perform manual counts of content processed. Vamosa Content Migrator is *VIA ready* and allows for the easy access of the relevant quality assurance and project or process management data through its intuitive user interface.

The VIA documentation also includes full exemplars of project plans to allow the fast path commencement of a VIA controlled migration project using the Vamosa Content Migrator toolset.

Archived

A

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described in this appendix.

Locating the Web material

The Web material that is associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246792

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the redbook form number, SG246792.

© Copyright IBM Corp. 2006. All rights reserved.

Using the Web material

The additional Web material that accompanies this redbook includes the files shown in the following tables.

Table 15-1 Installation product documentation

File name	Subfile	Description
WebContentManagement-2-5-0-InstallationGuide.pdf		Product documentation covering installation for IBM Workplace Web Content Management 2.5 and 5.1

Table 15-2 Resources for building the River Bend site

File name	Subfile	Description
Resources_for_RiverBend Website.zip		Top level wrapper for all files and code samples

Table 15-3 Customization examples

File name	Subfile	Description
Material_for customization.zip		Top level wrapper for all files and code samples
	SimpleDisplayPortlet.zip	RAD Project Interchange file of the sample built in the Chapter 12, “Working with the application programming interface” on page 469. (See specifically, 12.5.1, “Developing a simple Portlet” on page 501.)
	ilwwcm-custom-template-portlet.war	Customizable Template Portlet
	CustomJSP.ZIP	Additional files that are used in the Customizable Template Portlet example

Archived

File name	Subfile	Description
	utils.zip	Additional files that are used in the Customizable Template Portlet example

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zipped file into this folder.

Archived

Archived

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 about ordering these publications, see “How to get IBM Redbooks” on page 622. Note that some of the documents referenced here might be available in softcopy only.

- *IBM WebSphere Portal for Multiplatforms V5.1 Handbook*, SG24-6689

Online resources

These Web sites and URLs are also relevant as further information sources:

- Download full Product documentation (Information Center):

  http://www.lotus.com/ldd/notesua.nsf

- Workplace Web Content Management 2.5 Information Center.

  http://www-10.lotus.com/ldd/notesua.nsf/find/wcm25

- Workplace Web Content Management 5.1 Information Center.

  http://www-10.lotus.com/ldd/notesua.nsf/find/wcm51

- WebSphere Portal 5.1 (Information Center):

  http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/index.html

- IBM Workplace Web Content Management ibm.com Home page:

  http://www.ibm.com/software/workplace/webcontentmanagement

- WWCM Support Gateway:

  http://www.ibm.com/software/lotus/support/workplace-webcontent-mgmt/support.html

- Workplace Web Content Management Forum on developerWorks:

  http://www.ibm.com/developerworks/forums/dw_wpforums.jsp

Archived

© Copyright IBM Corp. 2006. All rights reserved.

- IBM Workplace Solutions Catalog (for the Customizable Template Portlet and Editor Integration API)

  http://catalog.lotus.com

- There are many articles on Workplace Web Content Management in developerWorks:

  http://www.ibm.com/developerworks/

- Education Offerings from IBM:

  http://www.lotus.com/services/education.nsf/wdocs/educationhomepage

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

Archived

Index

Numerics
2-phase linkings 421

A
access rights within security model 158
Access Web Content Management 257
accessing unrestricted resources 585
action, scheduled move 272
actionPerformed method 502, 521, 562
adding
  button 559
  Content Viewer Portlets 337
  groups to WebSphere Member Manager 578, 596
  left navigator 305
  users to WebSphere Member Manager 578, 596
administrator password 83
administrator user ID 83
Administrators group, migrating 578
advanced cache 349, 359, 374
advantages, design 54, 56, 58, 61
alerts
  e-mail 112
ALL USERS group 579
alternate rendering concepts 46
anonymous user 580
any organization 206
AO Foundation 206
  Advanced search 245
  example 220
  intuitive step 220
  navigation 235
  Surgery Module 214
API 12, 37, 39, 46, 167, 430, 469, 531
application program interface. *See* API.
applying standard templates when migrating 602
approach for migration 602
approval Workflow 271
AptrixCmpnt context 176, 278, 370, 538
AptrixLibCmpnt name 175, 287, 369, 455, 529
aptrixsearch.properties file 328
architectural items and components 154
as-is migration 602
assigning
  e-mail action to Reject stage 274
  expire action to the Expired stage 272
  scheduled move and publish actions to the Published stage 273
assigning content item as default page 291
attaching to a remote server 497
attribute
  lastname 577
attribute formatting 553
authoring 48, 65
  IBM Workplace Web Content Management 2.5 137
Authoring Portlet 15, 36, 41, 49, 59, 74–75, 127, 136, 147–148, 152, 258, 323, 482, 574, 581
  configuration changes 49, 63
  content item 323
  current content object 323
  development section 155
  displayed document 147
  future release 148
Authoring Template 15, 154, 161, 177, 197, 227, 233, 268, 274, 276, 283, 341, 343–345, 483, 536
  content item 345
  image components fields 190
  pairing with presentation template 283
  related field 185
  text component 490
Authoring to Staging 67
autoLoad 116, 327, 361
automated rendering 119

B
base example of a good information architecture 201
base server installation 73
basic API development sample 501
basic caching 349, 359
basic mail routing configuration 112
beginning the implementation 414
benefits of using IBM Workplace Web Content Management 12

© Copyright IBM Corp. 2006. All rights reserved.

breadcrumb 170
breadcrumb navigator 310–311
broken links 170
building
  category 266
  Site Framework 259
  taxonomy 266
  Web site 256
  Workflow 268
bulk resource import 470
Bulletin Board 264, 536, 540
  configuration 543
  menu configuration 542
  Portlet 543
business case for migration 601
business goals 217
business stakeholders 249
business-critical information 7

C

cache 350
cache events timeline 378
cache recommendations for components types 373
cache size 358
cache strategy, documenting 376
cache type 358, 377
cache, default 357
cacher 115
cacher module 116, 364
cacher options 117
caching 114, 349, 359, 368
  choosing between strategies 351
  distributed map 385
  implementing 353
  options 380
  paradigms 350
  performance impacts 371
  servlet 383
  strategy 349, 351–352, 371, 379
carriage return and line feed (CRLF) 570
category 164–165, 228, 266–267, 483
category management 164
category menu 319–320
changing the connect.cfg file 446
chapters in this book, overview 26
choosing a caching strategy 351
Cloudscape 42, 51, 73–74, 129, 567, 587
Column Table 276, 528
  breadcrumb navigator 311
  HTML Quick Search form 334
  image library references 288
  Left Navigator 308
  Tools Navigator 326
commands
  XMLAccess 417, 575
common errors and resolutions 582
communications 250
component architecture 34
component catalog 376
component library 152, 155, 175, 177, 285, 293, 478
  image components 191
component sub-interfaces 482
component types 149
componentization 149, 151
config mode 415
configuration
  post installation 112
configure
  cacher options 117
  custom UI Portlet 543, 550
  DB2 Connect client on WebSphere Portal and IBM Workplace Web Content Management server 79
  default cache 357
  IBM Workplace Web Content Management 77
  IBM Workplace Web Content Management 2.5 137
  IBM Workplace Web Content Management cache policy 391
  Java Edition
    e-mail routing 112
    fatal e-mail alerts 112
  library 341
  Local Rendering Portlet 103
  parameters 573
  Personalized List Portlet 451
  Portal Document Manager
    documents 341
    folder 341
    library 341
  Portlet 109, 545
  Portlet in Personalization demo 464
  portlets 338
  WebSphere Portal 5.1.01 446
configuring
  dynamic cache 385

Archived

confirming configuration 86
connect tags 368
connect.cfg file 91, 115, 327, 358, 446, 592
advanced cache configuration 360
basic cache configuration 359
default module property 361
ModuleConfig section 116
connection error 572
considerations for reusability 151
content 6, 143, 162, 169, 448, 479, 489
editing considerations 146
rendering 43, 45
content author 8, 12, 14, 28, 42, 48, 63, 72, 143, 146, 154, 160, 222, 231, 249, 273, 408
content component 104, 150, 370, 380, 478, 489
content component container 478
content component interface 481
content creation 3, 8–10, 63, 154, 161, 470
content custom UI 536, 543, 550
content document 144–145, 214, 233, 408, 436, 479, 522, 562, 592
content item 152–154, 177, 228, 285, 292, 430, 554, 581
content item, assigning as the default content page 291
content layer 150
content Library 152, 289
content management
importance 4
content management within the context of applications 5
content model 433
content owner 12, 17–18, 154, 217, 219–220
content page 151, 172, 203, 223, 288, 414, 425, 430, 478
multiple thousands 203
related information 314
search queries 333
third column 321
content viewer
Portlet 464
Content Viewer Portlets 337
creating and copying 336
content, migrating 599
content, testing 289
content-centric items 154
context 5
copying
Content Viewer Portlets 336
Local Rendering Portlet 96
River Bend Authoring Template 276
crawling
custom approach 436
regular Web sites mode 436
WebSphere Portal site mode 433
creating
Actions 272
approval Workflow 271
Authoring Template 274, 343
breadcrumb navigator 310
Bulletin Board page with portlets 540
categories for the site taxonomy 267
content 288–289, 293, 461
content item 291
Content Viewer Portlets 336
content with the River Bend Portal Document Manager Authoring Template 345
database alias 80
Development Express Workflow 270
documents 342
Edit Document Navigator 323
e-mail action 273
folder 342
HTML component to format the site area menu 317
HTML components to specify bold and plain text formatting 308
HTML Quick Search Form 333
HTML Search Form 331
IBM Workplace Web Content Management content using Profile 461
IBM Workplace Web Content Management database on DB2 server 78
image components 285
image library components 285
intranet Workflow stages 268
Left Navigator 305
library 341
menu using categories 319
menus 314
Personalization Rule 449
Portal Page 337
Portal Page Layout presentation template 334
presentation templates 276–277
Profile Rule 457
Publish stage 269
Related Information Menu 321
scheduled move action 272

Archived

search results template 330
site 259
site area menu component 314
site areas 260
Site Map Navigator 312
site map navigator 312
site taxonomy 266
Tools Navigator 325
two-column table presentation template 279
two-column table search presentation template 281
Unrestricted Resources 584
creating content 3, 8–10, 63, 154, 161, 470
custom caching with connect tags 368
custom JSP 544, 557
custom Portlet 411, 418
custom site 471
Customizable Template Portlet 30, 147, 535, 553, 562, 618
customization 561

D

data caching 362
data format 570
data modeling, importance 14
data transfer 570
verifying 571
data transfer task 570
database
administrator password 83
administrator user ID 83
URL 84
database alias 80
database platform 35, 77
DB2 Server 86
DB2 software 13
default cache 356
default content 103, 288, 428, 480, 516, 534
activated site area 181
default content pages 316
default JSP 550
defining the information architecture 200
delivery definitions for migration 607
demilitarized zone (DMZ) 66
deploying 531
considerations 530
Portlet 95, 107
Description field 289, 558
design advantages 54, 56, 58, 61
design approach 249
design components 174
design disadvantages 55–56, 59, 62
designing
Portlets 486
determining
components of the Web site 229
developing a caching strategy 371
development 47, 65
development components 174
Development Express Workflow 270
direct access 54, 56, 201
IBM Workplace Web Content Management server 54, 56
direct feature links 241
disabling User Migration 597
disadvantages, design 55–56, 59, 62
display
site map navigator in the site index 313
Display Title 152, 275, 370, 538
distinguished name 583
distributed map 385
document 476
parent site area 522
Security section 592
Document ID 477
document ID 425, 477, 520–521, 564
Document Manager 192
document type 177, 201, 228, 477
documenting
cache strategy 376
documents 342
doEdit method 502, 517
doView method 502, 522, 534
dynamic cache 348
configuring 385
distributed map 385
installing 385
servlet caching 383
dynamic cache monitor
installing 386
dynamically generated links 170

E

e.printStackTrace file 476, 480, 563
edge component 384, 400
Edge Side Include 395

Edit Document Navigator 323
edit document navigator component 324
edit mode 415
edit mode JSP 520
editable component sub-interfaces 478
editing
  aptrixsearch.properties file 328
  challenges 146
  connect.cfg file 327
  Customizable Template Portlet 147
  separate content and presentation 146
  User Migration Mapping file 594
  wpconfig.properties file 588
editing considerations 146
e-mail 112–113
e-mail action 273
e-mail configuration for fatal mail alerts 112
e-mail, fatal alerts 112
embedded hyperlinks 243
enablement 250
enabling
  cacher 115
  edge component 400
  IBM Workplace Web Content Management Cloudscape 74
  IBM Workplace Web Content Management Search Module 327
  Portal Personalization 446
  servlet caching on portal server 386
  the correct caching module 116
  the correct caching subsystem 115
  user migration 590
enhanced migration 602
ensuring valid data format 570
Enterprise Content Management 6
environments 63
error
  connection error 572
  examining 571, 582
  failed to start database 572
  No schema exists 572
  resolution 571, 582
  user migration 581
ESI. *See* Edge Side Include
example infrastructure 63
exception handling format 605
expiration parameters 356
expiry date 272, 553, 604
explicit mapping approach 414
exporting
  site to HTML or XML 471
extending Web site with JSP support 485
external products 440
external workflow applications 471

F

failed to start database error 572
feature links 241
field names 558
file
  aptrixsearch.properties 328
  connect.cfg 91, 115, 327, 358, 446, 592
  connect.cfg file 327
  e.printStackTrace 476, 480, 563
  linkBuf.append 563
  migrationUsers.properties 83, 577, 583–584, 594
  migrationUsers.xml 578
  time.jsp 528
  utils.zip 559
  wpconfig.properties 570, 588
File Resource component 188
file system data 568–569, 587–588
files needed by IBM Support Desk 573, 584
fix
  PK02213 92
fixes 87, 127
fixpack
  PQ98531 93
fixpacks 127
formatting HTML components 308
formatting the Navigator 308
fully qualified distinguished name 583

G

generating
  User Migration Mapping file 593
goals 217

H

high performance (caching) options 380
home page
  initial layout 199
href 175, 304, 561
HTML 45, 47, 170, 175, 181, 189, 228, 251, 305, 308, 314, 325, 373, 430, 470–471, 544, 610, 615

Archived

HTML component 317
HTML components 308
HTML formatting components 309
HTML Quick Search Form 333–334
HTML Search Form 331–332

I

IBM developerWorks 203
IBM HTTP Server 45, 54, 56, 115, 351, 440
  friendly message page 66
  mod_rewrite module 440
  recent version 120
  required resources 122
  single set 61
  static files 115
IBM Redbook team
  image component 292
IBM Redbook Web site 622
IBM Support Desk 573, 584
IBM Workplace Collaboration Services 24
IBM Workplace Web Content Management 8–10
  activity 54–55
  administration 56, 62
  administrator 154, 156
  advanced caching 349
  API 46, 376, 432, 436, 470–471, 473
  application 34–35, 72, 122, 129
  authoring portlet 36
  backend 425
  benefits of using 12
  cache policy 391
  cacher 115
  caching 349, 381
  caching performance impact 371
  capability 530
  changes from older version 41
  changes to installation 71
  code 23
  component 156, 362
  component library 175
  configuration 86, 348, 361
  configuration change 87
  configuration file 72
  configuration, changing 82
  configuration, confirming 86
  configure DB2 Connect client 79
  configure with DB2 Universal Database 77
  connect.cfg file 91, 118
content repository 35
content server 35
CPU entitlement 23
creating content 461
creating database on DB2 server 78
data 77, 566, 569
database 73, 81
  administrator password 83
  administrator user ID 83
  URL 84
definitions 152
document 408, 436–437
document ID 427
document title 436
EAR file 72
enabling with Cloudscape 74
engine 114, 368
environment 36–37, 47, 70, 230
feature 5, 22
fields 152
file system data 588
fix 93
fix PK02213 92
FixPack PQ98531 93
framework 348
function 155
functional differences between versions 126
functionality 470
functions 152
implementation 376
improving performance 348
Information Center 85, 118, 127, 361, 363
infrastructure 53
installation changes 71
installing 590
  API Java file 496
instance 349
integrating with WebSphere Portal 57
integration 446
introduction to 9
item 576, 578
item navigator 152
items 152, 154
JavaServer Pages tags 40
known limitations 573
license 23, 25, 41
Local Rendering Portlet 37
Local Rendering Server 37
location 231

Archived

log 113
logging 114
logs 87
migrating content 587, 599–605, 607–608
navigation 439
navigator 436
new features 14
node 531
object 349, 531
ODC editors 127
overview 10
page 149
parameter 570, 589
parameters 573
performance 348, 375
personalization rule 447
planning, site area 223
Portal Personalization 128
Portal search considerations 435
Portlet 57, 501
Portlet application 503
positioning 11
pre-rendered URLs 120
pre-rendering 28, 44, 351–352, 364, 380
previous version 30, 565
  file system data folder 569
  group names 582
  Lotus Workplace Web Content Management users 573
  migrating data 566
  predefined group 579
  predefined user 580
  Web site 30
process 46
processor 61
product 3, 10
project 16–17
public API 39, 46
Remote Rendering Portlet 38
removing 122
render 43, 45
rendering Portlet 37, 438
Rendering Portlets 95
repository 72, 362, 408
request 44
resources 585
sample 538
sample Workflow Portlet configuration 545
sample Workflow Portlet customization 561
search 228
search collection 435
Search Module 327, 430
separating 60
server 35, 46, 71, 157, 348, 431, 441, 473, 491
server administrator 156
server components 71
server crash 67
server tier 63
ship 114
simple 37, 46
site 172, 349, 351
site framework 223
software 44
stand-alone environment 43
stand-alone license 22
Standard Edition 9, 24
stores resource 585
system 10, 14
table 87
tag 351
tier 60
upgrading 567
user 72
using 21
Version 2.5 11, 24, 126, 129, 566, 574
  authoring 137
  configuring 137
  differences 126
  installing 128–130
  license 24
  overview 126
  running 126
Version 5.1 10, 27, 35, 70, 87, 126, 197, 374, 503, 566
  component architecture 34
  differences 126
  installing 27, 87, 92
  key components 35
  known issues with pre-rendering 119
Version 5.1.0.1 374
WebSphere Portal 23, 37
Workflow 50, 152
identifies the properties 361
identifying performance impacts of caching 371
iFixes 127
image component 163, 175, 190, 275, 285, 292
  width fields 287
image components, creating 285

Archived

image components, referencing 285
image fields 177
image library components 285
implementation
beginning 414
implementing
caching strategy 353
custom display Portlet 418
link lanagement 422
links, 2-phase linking 421
pre-rendering strategy 353
importing
from HTML 470
from third-party 471
improved configuration technique 72
improving performance 348
Including rule in JSP 467
including rule on presentation template 453
information architecture 3, 14, 17, 170, 199, 206, 215, 372
additional features 204
addresses key principles 28
critical component 232
elegant solution 214
exceptional example 214
good example 205
information architecture layer 150
information architecture, defining 200
Information Center 85, 118, 127
information design 232
information within the context of Web content 4
initworkspace tag 488
installation changes 71
installing
dynamic cache 385
dynamic cache monitor 386
Edge Side Include 395
IBM Workplace Web Content Management 73, 530, 590
IBM Workplace Web Content Management 2.5 128–130
IBM Workplace Web Content Management 5.1 27, 87, 92
IBM Workplace Web Content Management API Java files 496
overview 70
post installation configuration 112
PQ98531 fix 93
Rational Application Developer 6.0 492
Remote Rendering Portlet 107
Rendering Portlets 95
WebSphere Portal 5.1.01 446
integrating
IBM Workplace Web Content Management 446
WebSphere Portal and IBM Workplace Web Content Management 57
intranet 268
intranet portal 216, 536
item
labeled version 153
item and component security 153, 158
iterator 479, 485, 551

J

J2EE application 40
Java 158
Java code 23, 40, 42, 474
Java Content Repository 445–446, 448
JavaScript array 423, 563
JCR. *See* Java Content Repository.
JSP 40, 192, 278, 280, 354, 364, 407, 423, 467, 470, 485–487, 490, 526, 528, 544, 550
JSP components 526
JSP support 485
JSP tag library 487, 490

K

key words 228

L

labeled version, content item 153
lastname attribute 577
law of diminishing returns 616
layout
approach 249
portal page 407
LDAP 42, 70, 72, 92, 129, 246, 445, 447, 583
LDAP users and groups, migrating 579
left navigator 307
left navigator component 309
Library component 104, 158, 286, 368, 376, 470, 481, 490, 554
library, creating and configuring 341
license 22–23, 25, 41
IBM Workplace Web Content Management 2.5 24

Archived

line feed (LF) 570
link integrity 427
link management 418–419, 425
  implementing 422
linkBuf.append file 563
linking process 424
links
  2-phase linking 421
  broken 170
  dynamically generated 170
Links section 106
local rendering
  Portlet 15, 37, 41, 74, 95–96, 103, 336, 364–365, 408, 414–415, 437
  server 37
log 113
log files, using 581
logging levels 113
logs 87
Lotus Workplace 37, 41, 126
Lotus Workplace Web Content Management 573

M

mail routing configuration 112
making the implementation more functional 416
Management API 432
managing content 5
manual rendering 119
manual versus automated rendering 119
manually running the pre-rendering engine 119
mapping file 577, 593
master configuration 90, 389, 591
Member Fixer utility 580
menu component 170, 182, 315, 321, 370, 481, 536
menu form 317
menus, creating 314
  creating
    menus 238
MENU-Site Area 176, 315, 491
metadata 161–162, 164, 223, 225–226, 228, 266, 371, 376–377, 449, 602, 604, 611, 613
method
  actionPerformed 502, 521, 562
migrating 567
  Administrators group 578
  applying standard templates 602
  approach and toolset 602
  as-is migration 602
  business case 601
  content 599–605, 607–608
  data 587
  data from previous version 566
  delivery definitions 607
  determining source classifications 603
  determining target definition 604
  enhanced migration 602
  exception handling format 605
  groups 573
  law of diminishing returns 616
  LDAP users and groups 579
  quality assurance requirements 607
  restructured migration 602
  roadmap 600–605, 607–608
  rules 600
  standardized migration 602
  transformation requirements 605
  undertaking 608
  user and group 581
  User Migration Mapping file 577
  users 573–574, 591
  users and groups 576
  using a user migration mapping file 593
  verifying 581
migration
  user and group 581
  verifying 581, 595
migrationUsers.properties file 577, 583–584, 594
migrationUsers.properties file, modifying 83
migrationUsers.xml file 578
Model SPI 417
modifying
  presentation template for the Search page 329
  wpconfig.properties file 83
modular approach to using components 151
multilingual 425
multilingual and personalized content display 425
multilingual content display Portlet 534
multiple Portal pages 439
multiple rendering Portlets 41

N

Name field 152, 259
navigational tools 170
Navigator 177, 308
no schema exists error 572

Archived

NullPointerException 571

O

Objectives and Key Success Factors 218
ODC (Rich Text) editors 127
overview of chapters 26

P

page design 15, 141, 149, 228, 478
parameters 573
parent site area documents 522
performance 348, 375
  improving 348
personalization demo 457
personalization essentials 444
Personalization Rule 449
personalization rule 374, 443, 447
personalized content 425
placeholder tag 181, 307, 317, 370
Planning the site framework and site areas 223
populating the JCR with existing content information 448
Portal Document Manager 127, 192, 196, 343–345
  Authoring Template 343
  Authoring Template, creating 343
  configure 341
  integration 127, 196, 341
  integration for Version 5.1 196, 341
  pairing the Authoring Template and the presentation template 344
  path reference 193
Portal page 43, 58, 61, 106, 334, 375, 404, 407, 415, 418–419, 433, 450–451, 543, 564
Portal Page Layout 334
Portal Personalization 128, 446
Portal search
  considerations 435
Portlet 15, 23, 36–38, 45, 57, 73, 77, 147, 229, 258, 334, 337–338, 404, 433, 437, 445, 450, 464, 488, 491, 501, 503, 533, 538, 542
  content viewer 464
  placing copy on a page 99
  rendering 41
Portlet configuration 545
Portlet customization 561
Portlet Data Bean 521
Postbody field 175, 294
  image component coffeebeans 294
pre-rendered URLs 120
pre-rendering 28, 351, 353, 363, 380
pre-rendering engine, running manually 119
pre-rendering via HTTP Server 351, 365
pre-rendering via IBM Workplace Web Content Management 352, 364
presentation 143
presentation template 15, 46, 152, 155, 174, 232–233, 276–277, 279, 281, 283–284, 303–304, 307, 310–311, 322, 324, 326, 329, 334, 344, 371, 419, 432, 453, 483, 525, 538
  breadcrumb navigator component 310
  caching events timeline 376
  consistent design element 191
  document navigator component 324
  entry 377
  entry ID 377
  HTML Quick Search Form 334
  image components 285
  left navigator 307
  left navigator component 307
  new Component 344
  new component 344
  Portal Document Manager component 344
  referencing images 287
  same cache 379
  section 279
  stylesheet 303–304
  Tools Navigator Component 326
presentational layer 150
previous version 566
  file system data folder 569
  group names 582
  Lotus Workplace Web Content Management users 573
  migrating data 566
  predefined group 579
  predefined user 580
  Web site 30
Private (Trusted) Network 66
processing
  User Migration Mapping file 595
production 49, 66
Profile 161, 461
Profile Rule 457
property name 358
Publish stage 269
PUMA SPI 426

Archived

Q

quality assurance for migration 607
Quick Edit demo 548
Quick Search Form 333–334

R

Rational Application Developer 487
Rational Application Developer 6.0
  installing 492
Read-only mode 158
Really Simple Syndication 8, 46
rearrange the site area order 265
re-creating the resources 597
Redbooks Web site
  Contact us xxii
referencing
  breadcrumb navigator component in the presentation template 311
  category menu 319–320
  edit document navigator component in the presentation template 324
  HTML formatting components in the left navigator component 309
  HTML Quick Search Form in the presentation template 334
  HTML Search Form in the Search content item 332
  image component in a content item 292
  image components 285
  images in a presentation template 287
  left navigator in the presentation template 307
  menu format in the site area menu's component design 317
  new component in the presentation template 344
  related information menu in the Three Column Table presentation template 322
  site area menu in specified default content pages 316
  stylesheet in the presentation template 304
  Tools Navigator component in the presentation template 326
Related Information Menu 321
relative paths 119
remote DB2
  database 73
  server 70
Remote Rendering Portlet 15, 36, 38, 41, 95, 107
  configuration settings 117
  configure mode 109
  Credential section 110
  credentials 110
remote server, attaching to 497
removing
  IBM Workplace Web Content Management 122
RendererUser 118, 364
rendering 119
  content 43, 45, 478
rendering IBM Workplace Web Content Management content 43, 45
rendering options, overview 43
rendering Portlet 37, 438
Repository interface 474
representative skills 17
request.getHeader 426, 534
Resource Checker utility 586
Resource Description Framework 226
resources 121, 188
  re-creating 597
resources folder 121, 585
responsibilities 19
restructured migration 602
results of poor information architecture 215
return on investment (ROI) 441
re-usability 151
reusability 149, 151
reviewing
  User Migration Mapping file 594
Rich Text component 188
Rich Text editor 554
River Bend 28, 63, 66, 105, 141, 151–152, 165, 178, 202, 223, 230, 253, 255, 259, 265, 267, 276, 345, 412, 525, 530
  authoring 36
  Authoring Template 276
  building the Web site, prerequisites 257
  Coffee 28, 65, 141, 202, 253, 255
  Colombia Coffee 295
  creating content 293
  default content page 292
  development 163, 274
  External News 276
  External News Authoring Template 274
  footer 280
  footer image component reference 278
  Home page 149
  infrastructure 64–65

Archived

logo 277
MENU-Site Area 184
navigation 202
news brewing 291
open the Web site 303
page overview 229
Portal Document Manager 343–344
Portlets 337
River Bend Portlets 338
site 28, 64, 67, 105, 141, 253–254
site area 184
site area menu 316
site example 146
site framework 170
Tea 155
Web site Categories taxonomy 289
Web site Category 266
Web site display 177
Web site displays content 184
Web site text component 188
River Bend Coffee and Tea Company. *See* River Bend.
roadmap to migration 600–605, 607–608
roles 17, 19
routing, e-mail 112
running
data transfer task 570
IBM Workplace Web Content Management 126
wpsconfig script 85

S

same server
published content 15
sample physical architectures 53
sample Rich Text editor extensions 554
sample Workflow 545, 561
saving the stylesheet as a file resource component 304
scheduled move action 272
Search content item 332
search criterion 165, 170, 225, 306, 315, 449
Search Form 331
Search Module 327, 430–431
Search Module Index 431
search product 440
search result 244, 276, 283, 391, 430
search results template 330
search rules 194
security 153, 158
access rights 158
delete access 159
edit access 158
live access 158
read access 158
Search Module 431
security model
Java edition 158
Security section 592
security settings
refining Workflow security through values in aptrixjpe.properties file 160
select e-mail 273
selected site area
document ID 521
separating
IBM Workplace Web Content Management and WebSphere Portal 60
separating content and presentation 143
session bean 517, 520, 550
setContext 489
setExplicitContext 488
setting proper expectations about working with this product 16
setting up Caching 356
setting up the Development Environment 491
settings section 111
site 172, 259
site acceptance 249
site area 118, 154, 165, 172, 199, 202, 223, 260–262, 291, 317, 364, 426, 453, 478, 521, 534
AO Foundation navigation 235
appropriate name 262
Authoring Template 233
children site areas 307
content paths 517
creating menu component 314
default content 118, 181, 307
default content page 288
default content pages 316
format content 276
hierarchal list 518
ID 475
individual designs 172
link order 261
main navigation 236
menu 316
menu format 314

Archived

menu specific 184
new content 316
new pairs 173
parent site areas 310, 518
rearranging order 265
sequential placement 262
site area 292
template mappings 285
top levels 202
site design 27–28, 151, 199
site framework 3, 154, 170, 177, 199, 223, 259, 380, 479–480
appropriate Authoring Template 177
children site area nodes 265
container 480
only root 173
same hierarchy 263
site areas 173
site index 313
site map
navigator 312–313
navigator component 312
Site Map Navigator 312
site map navigator 312–313
site navigators 234
site taxonomy 266–267
site visitor 29, 37, 42, 72, 169, 331, 333, 349–350, 581, 584
software bundles 24
source classifications for migration 603
stage
Publish 269
staging environment 48–49
functional elements 49, 64
performance testing 49
Staging to Production 67
stakeholders 249
stand-alone license 22
standardized migration 602
static pre-rendering 44, 114
String headerLang 534
string representation 520
stylesheet 106, 155, 276–277, 303–304, 348
supported browsers 126
supported JVMs 127
syndication 50, 67, 153, 157, 231
system programming interface 417
system-defined security level 159

T

target definition for migration 604
taxonomy 164, 193, 266–267, 484
technical background to seed-list indexing method 433
Technote
1201178 87
template catalog 376
template, search results 330
testing
database catalog 81
testing your site 289
text component 188, 275, 373, 482, 538
text field 162, 544
third-party products 440
Three Column Table 322
time to implement 441
time.jsp file 528
tool for managing large amounts of content 21
Tools Navigator 325
Tools Navigator Component 326
Tools Navigator component 326
toolset for migration 602
transferring 567
file system data 568, 587
transferring file system data 569
transformation requirements for migration 605
troubleshooting 571, 581
data transfer 571
user and group migration 581
two-column table presentation template 279
two-column table search presentation template 281
typical component types 149
typical responsibilities within the different roles 19

U

UI access group 579
uninstalling IBM Workplace Web Content Management 122
Unrestricted Resources 584
unrestricted resources, assessing 585
upgrading 567
use case scenarios 222, 470
user goals 217
user interface 7, 10, 36, 158, 494, 578
user migration 574
error 581
mapping file 575

Archived

process 575
User Migration Mapping file 577, 593–595
User Migration, disabling 597
user profile 485
User Repository 42
user-centric design 219
user-defined security level 159
user-friendly URLs 421
username component 194
utils.zip file 559

V

valid data format 570
Vamosa 600–601, 603, 607, 609–613, 615–616
Vamosa Content Migrator. *See* Vamosa.
Vamosa Implementation Architecture (VIA) 601
verification, exception, and correction document (VEC) 606
verifying
data transfer 571
migration 581, 595
version management 153, 155
view content in a Portlet 334
view mode JSP 524
views navigator 259
content library 291
image component 286
right side 290
site framework view 291
site management 292

W

WCM_API class 474
wcm_migration 571
wcmadmins group 73, 578
Administrators group 578
WcmDbLibrary 85
WcmDbName 84
WcmDbUrl 84
WcmDbUser / WcmDbPassword 84
Web content 3–4, 33, 46, 69, 97, 254, 258, 429, 489, 526, 533
Web content management 69
key concepts 3, 8, 19, 254
overview 6
process 12
Web content management process 12
Web page 478
fast rendering 49
left side 305
search query 333
Web server 9, 45, 122, 348
alias entry 367
cache settings 375
static HTML 45
Web site
appropriate places 21
basic level 223
customized design 155
design elements 158
determining components 229
different elements 14
different sections 225
extending 485
hierarchical structure 154
information architecture 216
left-hand 305
logical arrangement 234
overview of building 256
share information 218
site map 223
specific objectives 222
voice 391
WebSphere Application Server 40, 87, 132, 348, 386, 486, 574
administrative user 133
enterprise application 380
installation directory 93
Login screen 88
necessary configuration 497
necessary shared libraries 496
root directory 40
web-app directory 486
WebSphere Dynamic Cache Service 380
WebSphere Member Manager 15, 35, 44, 72, 257, 566–567, 576, 591
adding new users or groups 578, 596
correct access 87
existing Mary Jones user 576
migrating 577
migrating users 591
migrating using a mapping file 593
migrationUsers.xml 578
new Administrators group 578
user names 576
WebSphere Portal 42
WebSphere Portal 10, 45, 348, 433

Archived

administrative user 134
base installation 131
changes 91
configure DB2 Connect client 79
different physical server 70
entitlement 23
extensive knowledge 20
IBM Workplace Web Content Management 229
IBM Workplace Web Content Management solution 37
improved integration 10
installing and configuring 446
integrating with IBM Workplace Web Content Management 57
integration 404
limited IBM Workplace Web Content Management entitlement 23
logs 87
navigation 405
search 433
separating from IBM Workplace Web Content Management 60
Server 572
WebSphere Member Manager capabilities 42
Windows 2000 Advanced Server 129
wireframe demonstration 250
work products 251
Workflow 50, 160, 166, 230, 271, 484, 545, 561
actions 168
compared with syndication 49
security 160
stage 484
Stages 167
stages 268
working with JSP components 526
wpconfig.properties 72, 82, 567
data folder 569
database administrator user name values 572
IBM Workplace Web Content Management parameters 573
wpconfig.properties file 570, 588
wpsconfig script, running 85
ws.getById 480, 534

X

XML 471
XML configuration interface 417
XML structure 433
XML version 75, 391
XMLAccess 417, 575

Archived

Archived

Redbooks

Web Content Management for Portal 5.1 and IBM Workplace Web Content Management 2.5

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Archived

Archived

Archived

IBM Workplace Web Content Management

for Portal 5.1 and IBM Workplace Web Content Management 2.5

Installation, configuration, and deployment best practices

Building and managing a site

Advanced topics

IBM Workplace Web Content Management is a powerful and sophisticated Web content management product that is designed to accelerate development and delivery of critical business information. It enables end-to-end collaboration for content creation, approvals, management, retention, and publishing across Internet, intranet, extranet, and portal assets. Despite its power, it is remarkably easy to use.

This IBM Redbook provides a comprehensive approach to understanding and using IBM Workplace Web Content Management. It addresses the specific technical aspects of IBM Workplace Web Content Management and discusses its underlying architecture and features. It also recommends deployment scenarios and provides detailed discussions on how to best use the product to take advantage of built-in features and its extensible API.

To address the needs of a broad range of users, this book provides a basic tutorial and then builds upon this foundation to explore advanced topics and opportunities for customization. This book discusses key concepts and best practices for creating a accurate Information Architecture and site framework. It also addresses how to upgrade data from earlier versions of the product and how to migrate data from external Web content management systems.

SG24-6792-00 ISBN 0738494259

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