



IBM Content Manager OnDemand Guide

Wei-Dong Zhu

Jim Ilardi

Deborah Matamoros

Trina D Morgans

Paula Muir

Hassan A Shazly

Edward E Stonesifer

Vanessa T Stonesifer

Sebastian Welter



ECM

IBM®

Redbooks



International Technical Support Organization

IBM Content Manager OnDemand Guide

October 2015

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

Fifth Edition (October 2015)

This edition applies to IBM Content OnDemand Version 9.5.

© Copyright International Business Machines Corporation 2003, 2015. All rights reserved.

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

Contents

Notices	xi
Trademarks	xii
IBM Redbooks promotions	xiii
Preface	xv
Authors.....	xv
Acknowledgment.....	xvi
Now you can become a published author, too!	xviii
Comments welcome.....	xviii
Stay connected to IBM Redbooks	xviii
Summary of changes.....	xxi
October 2015, Fifth Edition	xxi
Part 1. Basic system concepts and design	1
 Chapter 1. Overview and concepts	3
1.1 Overview of Content Manager OnDemand.....	4
1.2 Content Manager OnDemand concepts	6
1.2.1 Background information of an example company.....	6
1.2.2 Reports and documents	6
1.2.3 Application, application group, folder, and cabinet.....	8
1.2.4 Indexing methods	9
1.3 Content Manager OnDemand server and its components	10
1.3.1 Library server and object server	10
1.3.2 Content Manager OnDemand server components	11
 enChoice	13
 Chapter 2. Setting up a Content Manager OnDemand instance.....	15
2.1 Introduction	16
2.2 Architecture and platform	16
2.2.1 Configuration consideration	17
2.2.2 Library server and object server functions	17
2.2.3 Choosing a platform	18
2.3 Implementing a Content Manager OnDemand instance on a multiplatform UNIX environment	19
2.3.1 Defining a single instance	19
2.3.2 Starting and connecting to the new instance	28
2.4 Implementing a Content Manager OnDemand instance on IBM i.....	29
2.4.1 Configuring the instance	30
2.4.2 Changing an instance configuration	32
2.4.3 Starting and stopping servers	33
2.5 Implementing a Content Manager OnDemand instance on z/OS.....	36
2.5.1 Installation overview	37
2.5.2 Creating an instance on z/OS.....	38
2.5.3 Starting and verifying the new instance	43
 Chapter 3. Administration	45

3.1 Report administration	46
3.1.1 Storage sets	46
3.1.2 Application groups	47
3.1.3 Applications.....	51
3.1.4 Folders	54
3.1.5 Cabinets	58
3.1.6 The report wizard	58
3.2 User and group administration	66
3.2.1 User types, authorities, and functions.....	67
3.2.2 System administration.....	68
3.3 Content Manager OnDemand XML Batch Administration.....	72
3.3.1 Using the XML Batch Administration program	72
3.3.2 Special features of the XML batch program	74
3.3.3 Tips on using the ARSXML command	75
Chapter 4. Database structure	77
4.1 System control tables	78
4.2 Main data table structures.....	80
4.3 Relationship between tables when data is loaded	82
4.4 Search sequence	84
4.5 System log	85
4.6 Database creation and relationships on z/OS.....	85
4.6.1 System tables for Content Manager OnDemand z/OS.....	86
4.6.2 Data tables for Content Manager OnDemand z/OS	86
Chapter 5. Storage management	89
5.1 Content Manager OnDemand cache storage	90
5.2 IBM Tivoli Storage Manager for Multiplatforms.....	90
5.2.1 Tivoli Storage Manager overview	91
5.2.2 Configuring Content Manager OnDemand for Tivoli Storage Manager archive management.....	92
5.2.3 Content Manager OnDemand storage management	95
5.2.4 Storage set definition	96
5.2.5 Application group storage management	100
5.2.6 Advanced application group storage management.....	103
5.2.7 IBM System Storage Archive Manager.....	104
5.2.8 The arsmaint command	107
5.3 Object access method for z/OS	108
5.3.1 OAM components and SMS terminology	109
5.3.2 OAM configuration recommendations.....	110
5.3.3 Defining a storage set.....	113
5.3.4 Storing data in Virtual Storage Access Method datasets	117
5.4 Archive Storage Manager for IBM i	119
5.4.1 Migration policy	119
5.4.2 Application group storage management	126
5.4.3 Advanced application group storage management.....	128
Chapter 6. Security	131
6.1 Content Manager OnDemand security overview	132
6.2 Code security	133
6.2.1 Controlled environment.....	134
6.2.2 Code scanning	134
6.2.3 Quality assurance testing	134
6.3 Data separation.....	135

6.3.1	Multiple systems	135
6.3.2	Multiple object servers	135
6.3.3	Multiple archive servers	136
6.4	API access	137
6.5	Data security	138
6.5.1	Content Manager OnDemand object-owner model.	138
6.5.2	Administrative features	139
6.5.3	SQL macro support.	144
6.5.4	Annotations security	145
6.5.5	Securing access with ARSSTASH and the stash file	147
6.6	Data encryption	149
6.6.1	Encrypting data at rest	149
6.6.2	Encrypting data in motion: Secure communications.	149
6.7	Security exits	151
6.7.1	User security and permissions exits	152
6.7.2	Security systems other than SAF (z/OS only).	155
6.7.3	Unified logon exit (ARSPTGN): z/OS only	156
6.7.4	System log user exit	157
6.8	Summary	157
Part 2.	Data indexing, loading, retrieval, and expiration	159
Chapter 7. Indexing and loading 161		
7.1	Introduction	162
7.1.1	Loading and indexing files that were created on another system	163
7.1.2	Understanding input data types	163
7.1.3	Choosing an indexer	164
7.2	Getting started with PDF indexing	165
7.2.1	Limitations	166
7.3	Performance considerations	166
7.3.1	PDF fonts and output file size	166
7.3.2	Reducing output file size with PDF documents.	167
7.3.3	PDF indexing: Using PDF metadata	168
7.3.4	PDF indexing: Using the report wizard (graphical indexer)	168
7.3.5	PDF indexing: Using internal indexes (Page Piece Dictionary)	173
7.4	Getting started with ACIF indexing	173
7.4.1	Understanding the input data	175
7.4.2	The index file	177
7.4.3	Fully composed AFP input	178
7.5	OS/390 indexer on z/OS and AIX	179
7.6	OS/400 indexer on Content Manager OnDemand on IBM i	180
7.7	Getting started with XML Indexing	182
7.8	User exits	183
7.9	Additional references	183
Chapter 8. User clients 185		
8.1	Choosing the correct client for your implementation	186
8.1.1	Viewer options	186
8.1.2	Client infrastructure options	190
8.1.3	Client compatibility	191
8.2	Content Manager OnDemand Client options	193
8.2.1	IBM Content Navigator	193
8.2.2	Content Manager OnDemand Windows client	197
8.2.3	CICS Client	199

8.2.4	Integration with other Enterprise Content Manager products	201
8.2.5	Federated search with IBM Information Integrator	201
8.2.6	Integration with IBM FileNet P8.	201
8.3	Client API overview	202
8.3.1	Content Manager OnDemand Web Enablement Kit	202
8.3.2	Content Management Interoperability Services	204
8.3.3	Other client-based API options	204
Chapter 9. Data conversion	207
9.1	Overview of data conversion	208
9.1.1	Why convert data streams	208
9.1.2	When to convert data streams	208
9.1.3	How to convert the data	209
9.2	Generic Transform Interface	209
9.2.1	Overview	210
9.2.2	Configuration.	210
9.2.3	Basic implementation: Executable interface	211
9.2.4	V9.5 enhancement: Customizing values that are returned from ODWEK	212
9.2.5	V9.5 enhancement: Application Group and Application-specific XML	212
9.2.6	Advanced implementation: Custom Java interface	214
Chapter 10. Migration and expiring data and indexes	219
10.1	Introduction	220
10.2	Loading and storing the data.	220
10.2.1	Storing the report (document) data	221
10.2.2	Storing the index data	224
10.2.3	Storing the resource data	224
10.3	Configuring for migration and expiration	225
10.3.1	Migrating index data	225
10.3.2	Epiring data and indexes.	226
10.3.3	Epiring document data	227
10.3.4	Epiring annotations	229
10.4	Reloading data	229
10.5	Expiration processing on Multiplatforms and z/OS	230
10.5.1	Content Manager OnDemand expiration: ARSMAINT	230
10.5.2	Epiring indexes	230
10.5.3	Removing documents from the Tivoli Storage Manager archive	232
10.5.4	Storage Manager-based expiration (z/OS only)	236
10.6	Epiring data on Content Manager OnDemand for i	239
10.6.1	Content Manager OnDemand expiration	239
10.6.2	Storage Manager expiration	240
Chapter 11. Exits	241
11.1	Introduction to user exits	242
11.2	ACIF exits	242
11.2.1	New macro for user exits	243
11.2.2	Input record exit	243
11.2.3	Index record exit	244
11.2.4	Output record exit	246
11.2.5	Resource exit	247
11.2.6	Debugging input user exit programs	247
11.3	OS/390 indexer exits	248
11.3.1	ANYEXIT exit	249
11.3.2	INPEXIT exit	249

11.3.3 INDEXEXIT exit.....	249
11.4 System administration.....	250
11.4.1 System log exit for Multiplatforms.....	250
11.4.2 System log exit for z/OS	253
11.4.3 Print exit for Multiplatforms	258
11.5 Customized functions (Multiplatforms and z/OS only).....	263
11.5.1 User exit header file (arscxit.h)	264
11.5.2 Load exit	267
11.5.3 Report specifications archive definition exit	269
11.5.4 Table space creation exit	272
11.5.5 ARSYSPIN and sample APKACIF exit on z/OS	275
Part 3. Advanced system concepts and design.....	281
Chapter 12. Scalability, reliability, and availability architectures	283
12.1 Scalability, reliability, and availability defined	284
12.2 Scaling a Content Manager OnDemand system.....	285
12.2.1 Vertical scalability	286
12.2.2 Horizontal scalability: Library server	288
12.2.3 Horizontal scalability: Multiple object servers	288
12.2.4 Horizontal and vertical scalability: Storage manager	289
12.2.5 Horizontal scalability: Multiple logical partitions and systems.....	290
12.2.6 Multiple server configuration rules	291
12.3 High availability	292
12.3.1 Redundant systems: All platforms.....	292
12.3.2 Multiple LPAR sysplex: z/OS	293
12.3.3 High availability: IBM i.....	294
12.3.4 Horizontal and vertical scalability summary	294
Chapter 13. Performance	297
13.1 Tuning Content Manager OnDemand to enhance performance.....	298
13.1.1 Content Manager OnDemand configuration	298
13.1.2 System logging	298
13.1.3 System management	299
13.1.4 Storage management	299
13.2 Data loading performance	300
13.2.1 Factors that affect the load performance	301
13.2.2 Recommendations	302
13.2.3 Load testing	303
13.3 Data retrieval performance	304
13.3.1 Data retrieval parameters	304
13.3.2 Factors that affect retrieval performance	305
13.3.3 Retrieval testing	306
13.3.4 System testing	307
13.4 Performance issues that are based on data type	307
13.4.1 PDF data.....	308
13.4.2 Line data	309
13.4.3 AFP data	310
13.4.4 Image data	312
Part 4. Enhancement options	313
Chapter 14. Report distribution	315
14.1 Introduction to Content Manager OnDemand Distribution Facility	316

14.1.1	What documents are needed	318
14.1.2	Who receives the documents	318
14.1.3	When the documents are retrieved and delivered	318
14.1.4	Where are they delivered	319
14.1.5	Cross-platform access	319
14.2	Defining the objects with the Administrator Client	319
14.2.1	Adding a recipient	320
14.2.2	Adding a recipient list	321
14.2.3	Adding a report ID	322
14.2.4	Adding a distribution	323
14.2.5	Adding a report bundle	325
14.3	Defining the objects by using batch administration	327
14.3.1	Recipient	327
14.3.2	Report ID	328
14.3.3	Distribution and report bundle	328
14.4	Customizable user exits	329
14.4.1	arsodfxa: Spool file dataset allocation attributes exit	329
14.4.2	arsodfxb: Banner, header, and trailer exit	329
14.4.3	arsodfxm: Bundle manifest exit	330
14.4.4	ODFProcessDist.java: Processed distribution exit	330
14.5	Status and monitor tool	331
Chapter 15.	Full text search	335
15.1	Introduction to full text search in Content Manager OnDemand	336
15.2	Full text search architecture in Content Manager OnDemand	336
15.2.1	Full Text Search Server	337
15.2.2	Index structure	337
15.2.3	Indexing document through FTS Exporter	338
15.2.4	Searching	338
15.3	Planning and installing FTS	338
15.3.1	Component overview	339
15.3.2	Installing the FTS Server	339
15.3.3	Operating system resources	340
15.3.4	Workload	340
15.3.5	Memory heap size	340
15.3.6	Index considerations	341
15.4	Configuring and operating full text search	342
15.4.1	Base configuration in Content Manager OnDemand	342
15.4.2	Configuration of the Full Text Search Exporter	344
15.5	Running the full text indexing process	346
15.5.1	Automatically indexing new data during the load	346
15.5.2	Indexing existing data through the arsdoc command	346
15.5.3	Indexing existing data through ODWEK	347
15.5.4	Running the FTS Exporter	347
15.6	Using full text search in Content Manager OnDemand clients	347
15.6.1	Syntax	348
15.6.2	Boolean searches	348
15.6.3	Wildcard searches and optional terms	348
15.6.4	Fuzzy and proximity searches	349
15.6.5	Weighted searches (boosting terms)	349
15.7	Troubleshooting tips	349
15.7.1	Content Manager OnDemand server log	349
15.7.2	Full Text Search Server log	350

15.7.3 Full Text Search Exporter trace	351
15.7.4 Authentication and FTS Exporter errors	351
Chapter 16. Enhanced Retention Management	353
16.1 Enhanced Retention Management overview	354
16.2 Configuring Enhanced Retention Management	354
16.2.1 Enabling Enhanced Retention Management	355
16.2.2 Identify the application groups	355
16.2.3 Specify the application group lockdown field	356
16.2.4 Enabling hold permission for the application group	357
16.2.5 Assigning hold permissions to users	358
16.2.6 Creating holds by using the Administrator Client	359
16.2.7 Configuring a folder to display that a document is held	360
16.3 Applying and releasing holds	361
16.3.1 Managing holds	361
16.3.2 Applying holds	361
16.3.3 Creating and removing custom holds	362
16.3.4 Search for hold documents	363
16.4 Enhanced Retention Management use cases	364
16.4.1 Ad hoc holds	364
16.4.2 Load holds	364
Chapter 17. Content Federation Services for Content Manager OnDemand and IBM Enterprise Records	365
17.1 Content Federation Services for Content Manager OnDemand and IBM Enterprise Records overview	366
17.2 Administration of Content Federation Services for Content Manager OnDemand for Enterprise Records	367
17.2.1 Enabling Content Federation Services for Content Manager OnDemand	367
17.2.2 Identify the application groups where Content Federation will be enabled	368
17.2.3 Specifying the application group field	369
17.2.4 Enable Content Federation permissions for the application group	370
17.2.5 Federating document metadata to Content Federation Services for Content Manager OnDemand	371
17.3 Content Federation Services for Content Manager OnDemand architecture	372
17.4 Deployment considerations	373
Part 5. Troubleshooting, hints, and techniques	375
Chapter 18. Troubleshooting and tracing	377
18.1 Troubleshooting common problems	378
18.1.1 Client issues	378
18.1.2 Indexing and loading issues	379
18.1.3 Content Manager OnDemand maintenance issues	381
18.1.4 Monitoring the main server task arsockd	383
18.1.5 Installation and migration issues	384
18.1.6 Common server messages	388
18.2 Information collection	389
18.2.1 Indexing or loading	390
18.2.2 Database	391
18.2.3 Tivoli Storage Manager	394
18.2.4 Content Manager OnDemand Client logon	394
18.2.5 Performance	395
18.2.6 ODWEK	396

18.2.7 Content Manager OnDemand server hangs or crashes	396
18.2.8 Exporting information to a local server	397
18.3 Content Manager OnDemand trace facility	399
18.3.1 Enabling the trace facility	400
18.3.2 Setting trace parameters	400
18.4 Other tracing options	402
18.4.1 ARSLOAD	402
18.4.2 MidServer trace (z/OS only)	403
18.4.3 ODWEK trace	403
18.4.4 TCP/IP packet trace	404
18.4.5 Language Environment (z/OS only)	405
18.4.6 ARSSUPPORT utility	405
18.4.7 ARSJESD	406
18.4.8 PDF Indexer trace	406
18.4.9 Trace resolver	407
18.4.10 Conclusion	407
Related publications	409
IBM Redbooks	409
Other publications	409
Online resources	410
Help from IBM	411

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

AIX®	IMS™	RACF®
CICS®	Language Environment®	Redbooks®
Daeja™	Lotus®	Redbooks (logo)  ®
DB2®	MVS™	System Storage®
DB2 Universal Database™	OMEGAMON®	System z®
developerWorks®	OS/390®	SystemMirror®
FileNet®	OS/400®	Tivoli®
IBM®	Power Systems™	WebSphere®
IBM z™	PowerHA®	z Systems™
IBM z Systems™	Print Services Facility™	z/OS®

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Evolution, and Inc. device are trademarks or registered trademarks of Kenexa, an IBM Company.

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

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

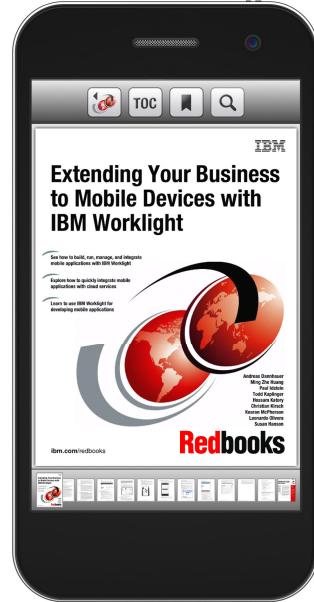
UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Find and read thousands of IBM Redbooks publications

- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the **Redbooks Mobile App**



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication provides a practical guide to the design, installation, configuration, and maintenance of IBM Content Manager OnDemand Version 9.5.

Content Manager OnDemand manages the high-volume storage and retrieval of electronic statements and provides efficient enterprise report management. Content Manager OnDemand transforms formatted computer output and printed reports, such as statements and invoices, into electronic information for easy report management. Content Manager OnDemand helps eliminate costly, high-volume print output by capturing, indexing, archiving, and presenting electronic information for improved customer service.

This publication covers the key areas of Content Manager OnDemand, some of which might not be known to the Content Manager OnDemand community or are misunderstood. The book covers various topics, including basic information in administration, database structure, storage management, and security. In addition, the book covers data indexing, loading, conversion, and expiration. Other topics include user exits, performance, retention management, records management, and many more.

Because many other resources are available that address subjects on different platforms, this publication is not intended as a comprehensive guide for Content Manager OnDemand. Rather, it is intended to complement the existing Content Manager OnDemand documentation and provide insight into the issues that might be encountered in the setup and use of Content Manager OnDemand. This book is intended for individuals who need to design, install, configure, and maintain Content Manager OnDemand.

Authors

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

Wei-Dong Zhu is a Content Management Project Leader with the ITSO at IBM US, California. She is a Certified Solution Designer for IBM Content Manager. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution (DMD). Her development work in one of the DMD solutions contributed to a first-time ever win for IBM of an Emmy award in 2005. Jackie joined IBM in 1996. She holds a Master of Science degree in Computer Science from the University of Southern California.

Jim Ilardi is a Consulting Client Solution Professional in Carmel, New York. Jim has over 30 years of experience in IT and over 18 years working with Content Manager OnDemand. Jim started with IBM in Lab Services installing many OnDemand systems around the world. Today, Jim works in Pre-Sales Technical Sales covering Enterprise Content Manager in the New York Area.

Deborah Matamoros is a Software Developer with Content Manager OnDemand. She has 26 years of development experience at IBM, with the last seven of those years in OnDemand. Her area of expertise is Report Distribution. Debbie holds a degree in Computer Science from the University of Oregon and currently resides in Park City, Utah.

Trina D Morgans is a Level 2 Support Engineer with Content Manager OnDemand for IBM z/OS® in West Chester, Pennsylvania. She has three years of experience with Content Manager OnDemand, and 30 years of experience in IBM DB2® for z/OS. She has worked at IBM for 12 years. Her areas of expertise include database, performance and tuning, installation, migration, and overall Content Manager OnDemand troubleshooting. She has co-authored books on DB2 data warehousing and referential integrity. Trina holds multiple degrees in both Computer Science and Technology.

Paula Muir is a Software Developer with Content Manager OnDemand for Multiplatforms in Boulder, Colorado. She has 20 years of experience with Content Manager OnDemand, and 15 years of experience in the data indexing field. She has worked at IBM for 19 years. Her areas of expertise include indexing and loading data, and AFP and PDF architecture. She has written extensively about using AFP Conversion and Indexing Facility (ACIF) to index line data and AFP documents, and indexing PDF documents. Paula holds a degree in Mathematics and Computer Science from the University of Illinois.

Hassan A Shazly is a Development and Quality Assurance (QA) Manager for Content Manager OnDemand. He has over 37 years of experience in Information Systems, the last 17 of which were at IBM. Hassan has written articles, submitted patents, and presented topics, including client/server technology, image processing, testing, and systems performance, at several conferences. Hassan holds a Ph.D. in Image Processing from the University of South Carolina, US.

Edward E Stonesifer is an Executive Technical Sales Specialist with Mid-Atlantic Enterprise Content Management Business Unit at IBM US. Ed has 37 years of experience in Information Technology and 23 years of experience with Content Manager OnDemand, specializing across all IBM eServer™ platforms. Ed has been instrumental in providing input to the product development team for enhancements to meet the business requirements for all Content Manager OnDemand clients. Ed has presented at numerous technical conferences and has authored technical white papers addressing Content Manager OnDemand.

Vanessa T Stonesifer is a Software Engineer with the Content Manager OnDemand development team at IBM US, with a current focus on performance and system integration testing. Vanessa joined IBM Software Group in 1999 as a Technical Leader and Consulting Services Specialist in the Enterprise Content Management Software Group Services division – Content Manager OnDemand. Vanessa has over 29 years of experience in Information Technology and with over 18 years of experience with Content Manager OnDemand. She was the Technical Team Lead for Content Management Lab Services for more than 12 years, responsible for solution implementations, performance optimization, and migrations for Content Manager OnDemand. Vanessa has also presented at multiple technical conferences and Content Manager OnDemand User Groups and has co-authored multiple Content Manager OnDemand IBM Redbooks publications.

Sebastian Welter is a technical professional for Enterprise Content Management and Analytics in IBM Germany. He is on the European Advisory Council for Content Manager OnDemand. He has 10 years of experience with Content Manager OnDemand on various platforms. His areas of expertise include large Enterprise Content Manager architectures, as well as analytics application in the space of text analytics.

Acknowledgment

Special thanks to **Hassan A Shazly** for his effort in bringing this book up to the IBM Content OnDemand Version 9.5 level.

Special thanks to the following people for their content contribution:

Ben Boltz is a Senior Software Engineer. He has 32 years of experience in the Software Industry. He has worked on Content Manager OnDemand for Multiplatforms for over 20 years.

Darrell Bryant joined IBM as a manufacturing engineer and worked as a Systems Engineer who specialized in S/36 and AS/400 systems. In 2000, Darrell joined the OnDemand team. He has performed a mix of activities, including services, education, support, and testing. Darrell is now the lead tester for OnDemand for i. He also develops and teaches workshops to clients and partners. He is the editor of the OnDemand Newsletter.

Nelson Chen is a Software Developer with Content Manager OnDemand. He has over 30 years of experience in software development, among them 27 years at IBM and last 20 years in OnDemand. His areas of expertise include ArsXML, Install, and Configurator.

Trang Kim Duong is a Software Developer with Content Manager OnDemand. Among her 17 years of experience in software development, the last 11 years were in Content Manager OnDemand. Trang's areas of expertise include workflow for Space Vehicle Design, Content Manager OnDemand Report Distribution, Exporter utility for Content Federation Services for Content Manager OnDemand (CFS-CMOD), and the Content Manager OnDemand back-end database component.

Hubert Hwang is a Software Developer with Content Manager OnDemand. He is a Certified Solutions Expert for Content Manager OnDemand with over 10 years of experience with the product. His areas of expertise include the Content Manager OnDemand Web Enablement Kit Java application programming interfaces (APIs), Content Navigator, and software test automation. He has extensive experience troubleshooting all aspects of the product. He has authored over 200 technotes on topics, such as migration, data collection, and troubleshooting guides, for Content Manager OnDemand.

Vicki Miller is a Senior Certified Client Technical Professional at IBM, working in the technology industry for 33 years with a focus on Enterprise Content Manager (ECM) since 1999. Her area of expertise in the realm of ECM is focused on solution sales consulting and technical account leadership that revolves around the management, processing, and analysis of any type of electronic content to help organizations optimize and protect their business. Vicki has spoken at IBM conferences on critical ECM topics, contributed to the development of technical publications, and led groups within IBM and client organizations to drive the enhancement ECM solutions and products.

Paula Muir is a Software Developer with Content Manager OnDemand for Multiplatforms in Boulder, Colorado. Her areas of expertise include indexing and loading data, and AFP and PDF architecture.

Nancy O'Brian started at IBM as an applications programmer, then transferred to a branch office where she performed her first Content Manager OnDemand (then known as R/DARS) implementation. After many more implementation service engagements, she joined the Content Manager OnDemand development team and continued to perform implementation services, training, support, testing, and technical writing. She currently focuses primarily on technical writing and testing.

Sandi Pond is a Software Developer with Content Manager OnDemand for Multiplatforms. She has 17 years of experience with Content Manager OnDemand, working in various areas of the development team. Her area of expertise is the OnDemand Web Enablement Kit (ODWEK).

Debbie Wagner is a Senior Software Engineer at IBM and has over 22 years of experience in content management, specifically, Content Manager OnDemand for Multiplatforms. Her areas

of expertise are graphical user interfaces (GUIs), product administration software, and document retrieval. She has written and co-authored multiple articles that relate to Content Manager OnDemand. Debbie is an IBM certified deployment professional for Content Manager OnDemand V9.0.

Additionally, special thanks to the following people who contributed their time and energy to assist us in producing the book:

Neil Parrott, Sr. Product Marketing Manager - Enterprise Report Management segment, IBM US

Stephanie M Wing, Worldwide Portfolio Marketing Manager, Enterprise Content Management, Content Manager OnDemand, IBM US

Whei-Jen Chen, Project Leader, International Technical Support Organization

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:
ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRredbooks>

- ▶ Follow us on Twitter:

<http://www.twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>

Summary of changes

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

Summary of Changes
for SG24-6915-04
for IBM Content Manager OnDemand Guide
as created or updated on May 4, 2016.

October 2015, Fifth Edition

This revision includes information about IBM Content Manager OnDemand Version 9.5.



Part 1

Basic system concepts and design

This part contains the following chapters:

- ▶ Chapter 1, “Overview and concepts” on page 3
- ▶ Chapter 2, “Setting up a Content Manager OnDemand instance” on page 15
- ▶ Chapter 3, “Administration” on page 45
- ▶ Chapter 4, “Database structure” on page 77
- ▶ Chapter 5, “Storage management” on page 89
- ▶ Chapter 6, “Security” on page 131



Overview and concepts

In this chapter, we provide an overview of the IBM Content Manager OnDemand (Content Manager OnDemand) system. We describe how Content Manager OnDemand manages reports and index data. We also provide information to help you better understand how Content Manager OnDemand works.

In this chapter, we cover the following topics:

- ▶ Overview of Content Manager OnDemand
- ▶ Content Manager OnDemand concepts
- ▶ Content Manager OnDemand server and its components

1.1 Overview of Content Manager OnDemand

To compete in today's global business environment, businesses must increase both the efficiency and effectiveness of their operations. Conflicting business requirements, such as increasing productivity while reducing costs and increasing personalization yet at the same time expanding to larger customer bases, can be achieved only through more streamlined and coordinated processes. Content Manager OnDemand helps address these issues by securely storing information and managing its delivery on demand whenever and wherever it is needed.

Content Manager OnDemand is the leading report archive system and it is used by thousands of organizations worldwide. The high scalability and high-speed information archiving and retrieving benefit any organization that requires instant access to information, hardcopy replacement, or long-term archival of data. A Content Manager OnDemand system can support small office environments and large enterprise installations with hundreds or thousands of system users. It can dramatically improve productivity and customer service in many businesses by providing fast access to information that is stored in the system.

Content Manager OnDemand is a robust report management system to perform the following tasks:

- ▶ *Capture*: Captures various data types from various sources through a batch capture system or interactively through custom-built interfaces.
- ▶ *Store*: Stores data for immediate retrieval.
- ▶ *Search*: Indexes data so that users can easily and quickly find the information.
- ▶ *Full Text Search*: Allows searching the full text of stored documents.
- ▶ *Integrate*: Enables organizations to integrate Content Manager OnDemand into their existing software stack by using components, such as OnDemand Web Enablement Kit (ODWEK). Organizations can also enable access through federated searches to other IBM Enterprise Content Management data and third-party products.
- ▶ *Display*: Supports multiple viewers for different data types, providing fast access for browsing and printing the retrieved data. For example, by using ready-for-use products, such as IBM Content Navigator, users can search and access Content Manager OnDemand, other IBM Enterprise Content Management data stores, and third-party products.
- ▶ *Distribute*: Distributes data to selected users (through email or print).
- ▶ *Manage*: Expires or archives data based on defined policies.
- ▶ *Archive*: Provides data archives online, near-line, or offline, enabling rapid archiving of data to the storage system.
- ▶ *Control*: Controls system and data access, allowing only authorized users to access specified data.

In summary, Content Manager OnDemand enables you to gain control of your information by providing access to your business' data, as needed, regardless of the size of the business or the hardware platform. Content Manager OnDemand improves your organization's bottom line by helping you become more efficient and responsive.

Figure 1-1 on page 5 presents an overview of the Content Manager OnDemand (OnDemand) system.

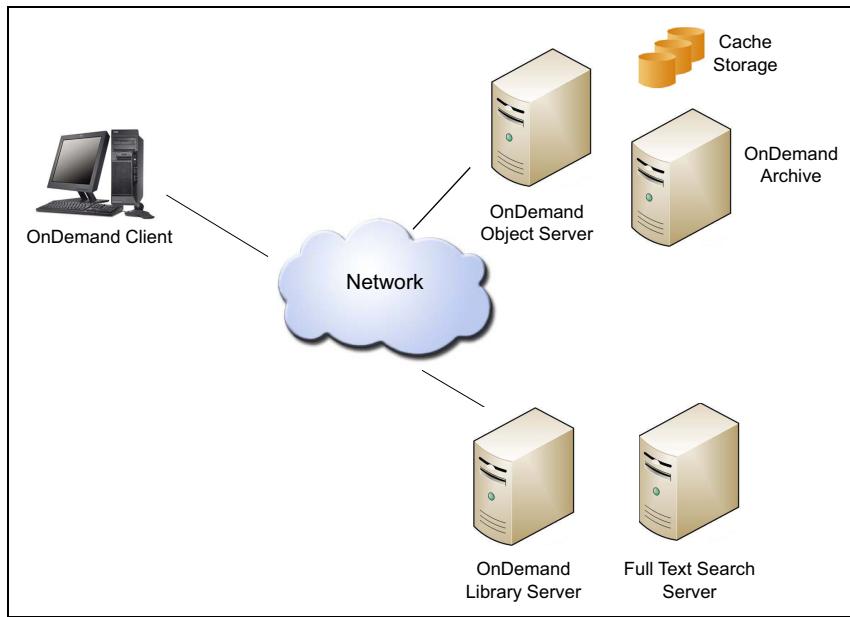


Figure 1-1 Content Manager OnDemand system overview

Content Manager OnDemand Client programs provide authorized users with high-speed access to the archived data that runs on the user devices (workstations) that are attached to the network and communicate with the Content Manager OnDemand servers.

A Content Manager OnDemand server consists of multiple components that can be installed on a single system or multiple systems. In all cases, the installation appears to the users as a single server. The installation is administered by the Content Manager OnDemand administrator as a single system.

The Content Manager OnDemand server includes the following components:

- ▶ A single library server: The library server manages a database that contains the information about the users of the system, and the reports and data that are stored on the system.
- ▶ One or more object servers: The object servers manage the data on disk or tape storage devices.
- ▶ One or more archive servers: The archive server stores the archived data objects. Depending on the operating system, the archive servers might be IBM Tivoli® Storage Manager, object access method (OAM), or Archive Storage Manager (ASM).

The library server and the object server can be packaged separately or as a single executable file.

Content Manager OnDemand Client programs

Content Manager OnDemand Client programs operate on various environments, including personal computers that are running on Windows, web browsers, and mobile devices. By using the client program, users can search for and retrieve reports that are stored on the system. Specifically, users can construct queries and search for reports, retrieve documents from Content Manager OnDemand, view, print, and fax copies or pages of documents, and attach electronic notes to the pages of a document.

Content Manager OnDemand servers manage control information and index data, store and retrieve documents and resource group files, and process query requests from Content Manager OnDemand Client programs. The documents can be on disk and tape storage volumes. New reports can be loaded into Content Manager OnDemand every day. This way, Content Manager OnDemand can retrieve the latest information that is generated by application programs.

When a user submits a query, the client program sends a search request to the Content Manager OnDemand library server. The library server returns a list of the documents that match the query to the user. When the user selects a document for viewing, the client program retrieves a copy of the document from the object server where the document is stored, opens a viewing window, and displays the document.

Full text search allows users to search the full content of stored documents. For example, users can perform wildcard searches, fuzzy (or similar) searches, proximity searches, and boolean searches.

Documents or reports can also be automatically distributed to users through email or network printers. The distributions can be scheduled to occur at the time that the data is loaded or at specific times during the day.

1.2 Content Manager OnDemand concepts

In this section, we examine basic concepts of Content Manager OnDemand:

- ▶ Report and document
- ▶ Application, application group, folder, and cabinet

1.2.1 Background information of an example company

As we examine these concepts, we use an example company. Our fictitious company is called AFinancial Co. AFinancial Co is one of the largest custodians of financial transactions in the world. It is one of the leaders in managing customer assets, providing financial services and foreign exchange services. It is also one of the leading credit card providers in the world.

The timely delivery of information and reports is fundamental to maintaining this leadership status. Products and services that provide real-time, online access to a customer's account and fund information are key to competitive differentiation and are key to customer retention. AFinancial Co's customers want personalized fund information, in various standard formats, which are delivered through both web and desktop interfaces.

1.2.2 Reports and documents

A *report* is one or more pages of data that is typically generated on a periodic basis by a computer software system. Content Manager OnDemand *documents* represent indexed groups of pages from a report. A Content Manager OnDemand document can be a logical section of a large report, such as an individual account statement within a report of thousands of statements. A Content Manager OnDemand document can also represent a physical portion of a large report. For example, if a large report does not contain logical groups of pages, such as transaction logs, Content Manager OnDemand can divide the report into groups of pages. The groups of pages are individually indexed and can be retrieved much more efficiently than the entire report.

Documents are identified (indexed) by date, with one or more other fields, such as customer name, customer number, or invoice number. A date is optional but highly recommended for optimizing document search performance.

Our example fictitious company, AFinancial Co, prints customer credit card statements monthly. This report, the customer credit card statements (Customer Statements), consists of thousands of individual customer statements. The company also prints transaction logs monthly. This second report, the transaction log (Transaction Report), contains thousands of customer transactions per month. The company must load these two reports into Content Manager OnDemand so that their data can be stored, then easily searched, retrieved, and viewed later. Let us look at how these two large reports might be broken up into individual Content Manager OnDemand documents.

Reports are “loaded” into the Content Manager OnDemand system. A Content Manager OnDemand application describes how the report will be divided into documents. Figure 1-2 on page 8 illustrates two reports, their associated Content Manager OnDemand applications, and documents. Let us look at how the associated applications divide the reports into Content Manager OnDemand documents.

The first report that we look at is the Customer Statements report. For this example, the report consists of 63,097 individual customer statements. An administrator can define a “*Statement application*” for this report that breaks up the report into logical documents. The Statement application uses the document indexing method to divide the report into documents that are based on customer name or customer number. Each statement in the report becomes a document in Content Manager OnDemand. Users can retrieve a statement by specifying the date and any combination of customer name and number.

Certain reports might not have a logical way of breaking up into individual documents. For example, the Transaction Report is not sorted by customer name or number. The report is generated based on the transactions of the day and time, and the customers that are associated with the transactions. In this case, we can break up the report into groups of pages. An administrator can define a “*Trans application*” for the report that contains lines of sorted transaction data. The Trans application uses the report indexing method to divide the report into documents. Each group of 100 pages in the report becomes a document in Content Manager OnDemand. Each group is indexed by using the first and last sorted transaction values (transaction date and number) that occur in the group. Users can retrieve the group of pages that contains a specific transaction number by specifying the date and the transaction number. Content Manager OnDemand retrieves the document that contains the value that is entered by the user.

To summarize this example, as shown in Figure 1-2 on page 8:

- ▶ Customer Statements report: Contains all customer statements for a month. Customer Statements documents: Each customer statement is a document.
- ▶ Transaction Report: Logs all transactions as they occur for a month. Transaction Report documents: Every 100 pages of the report are a document.

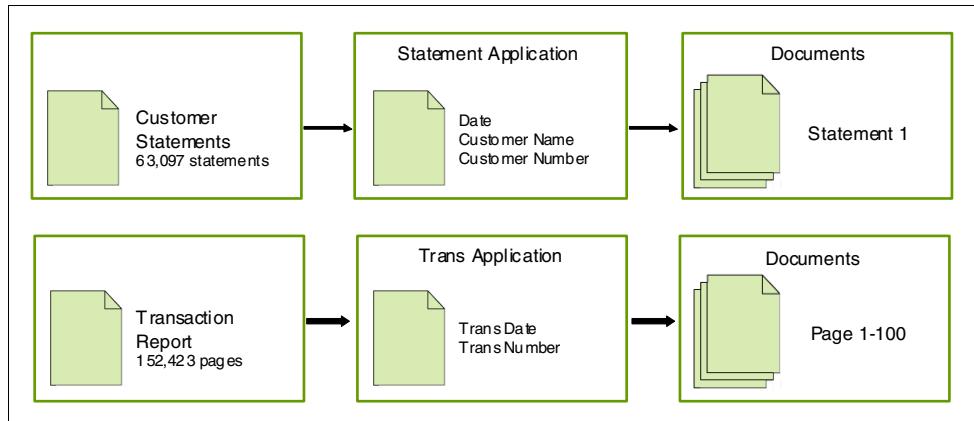


Figure 1-2 Reports and documents

1.2.3 Application, application group, folder, and cabinet

The terms *application*, *application group*, *folder*, and *cabinet* represent how Content Manager OnDemand stores, manages, distributes, retrieves, displays, and prints reports and index data. When you define a report or type of data to Content Manager OnDemand, an administrator must perform the following tasks:

- ▶ Create an application and assign the application to an application group.
- ▶ Create or update a folder to use the application group and application so that users can search for and retrieve documents.
- ▶ Optionally, create or update a cabinet. *Cabinets* are containers for collections of folders. They allow users to manage and navigate folders more easily.

Application

An *application* describes the physical characteristics of a report to Content Manager OnDemand. Typically, you define an application for each program that produces output to be stored in Content Manager OnDemand. The application includes information about the format of the data, the orientation of data on the page, the paper size, the record length, and the code page of the data. The application also includes parameters that the indexing program uses to locate and extract index data and processing instructions that Content Manager OnDemand uses to load index data in the database and documents on storage volumes.

Application group

An *application group* contains the storage management attributes of and index fields for the data that you load into Content Manager OnDemand. When you load a report into Content Manager OnDemand, you must identify the application group where Content Manager OnDemand loads the index data and stores the documents.

An application group is a collection of one or more Content Manager OnDemand applications with common indexing and storage management attributes. You typically group several related reports in an application group so that users can access the information that is contained in the reports with a single query. All of the applications in the application group must be indexed on one or more common fields, for example, customer name, account number, or date.

Folder

A *folder* is the user's way to query and retrieve data that is stored in Content Manager OnDemand. A folder provides users with a convenient way to locate related information that is stored in Content Manager OnDemand, regardless of the source of the information or how the data was prepared.

A folder allows an administrator to set up a common query panel for several application groups that might use different indexing schemes so that a user can retrieve the data with a single query. For example, a folder that is called "Customer Information" might contain customer credit card statements, checking and saving accounts, and mortgage payment information, which represent information that is stored in different application groups, which are defined by different applications, and created by different programs.

Cabinet

A *cabinet* is a container for folders. You can use cabinets to manage folders and enable users to navigate to folders more easily. A folder can belong to one or more cabinets.

Figure 1-3 summarizes these concepts.

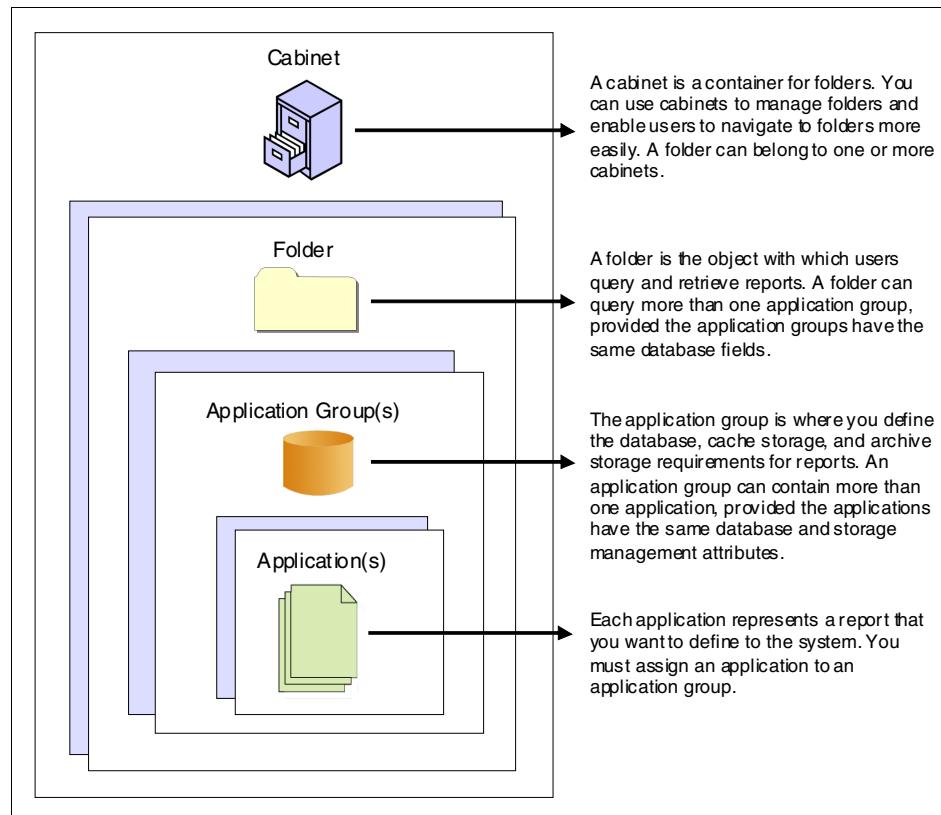


Figure 1-3 The concepts of cabinets, folders, application groups, and applications

1.2.4 Indexing methods

Content Manager OnDemand provides two methods of indexing data:

- ▶ Document indexing
- ▶ Report indexing

Document indexing

Document indexing is used for reports that contain logical items, such as customer name or number. Each of the items in a report can be individually indexed on values, such as account number, customer name, and balance. Content Manager OnDemand supports up to 128 index values per item. With document indexing, the user is not necessarily required to know about reports or report cycles to retrieve a document from Content Manager OnDemand.

Report indexing

Report indexing is used for reports that contain many pages of the same type of data, such as a transaction log. Each line in the report usually identifies a specific transaction, and it is not cost-effective to index each line. Content Manager OnDemand stores the report as groups of pages and indexes each group.

When reports include a sorted transaction value (for example, transaction date and number), Content Manager OnDemand can index the data on the transaction value. This indexing is done by extracting the beginning and ending transaction values for each group of pages and storing the values in the database. This type of indexing lets users retrieve a specific transaction value directly.

1.3 Content Manager OnDemand server and its components

On IBM z/OS® and Multiplatforms (MP) systems, the Content Manager OnDemand server can be implemented as a library server and one or more object servers that are on one or more nodes that are connected to a Internet Protocol network. For the Content Manager OnDemand system overview, see Figure 1-1 on page 5.

1.3.1 Library server and object server

A Content Manager OnDemand *library server* maintains two sets of database tables:

- ▶ The first set of database tables contains indexes about the reports that are stored in the Content Manager OnDemand Archive.
- ▶ The second set of database tables contains information about the objects that are defined to the system, such as users, groups, printers, application groups, applications, folders, cabinets, and storage sets.

The database manager provides the database engine and utilities to administer the database. The library server processes client logons, queries, and print requests and updates to the database. The major functions that run on the library server are the request manager, the database manager, and the server print manager.

A Content Manager OnDemand *object server* maintains documents on cache storage volumes and an ASM. ASMs, such as Tivoli Storage Manager on Multiplatform systems, OAM on z/OS systems, or ASM on IBM i systems, allow hierarchical storage management techniques to be applied to the stored documents. An object server loads data, retrieves documents, and expires documents. The major functions that run on an object server are the cache storage manager, data loading and maintenance programs, and optionally, the ASM.

The basic Content Manager OnDemand configuration is a library server and an object server on the same physical system or node. This single library or object server configuration supports the database functions and cache storage on one system. You can add an ASM to the single library or object server configuration to maintain documents on archive media.

On certain platforms, you can also configure your Content Manager OnDemand system with a library server on one node and one or more object servers on different nodes. These nodes can run the same or different operating systems. This configuration is known as a *distributed* library/object server system. The distributed library and object server configuration supports the caching of documents on different servers. You can add an ASM to one or more of the object servers to maintain documents on archive media that are attached to different servers. One of the key requirements in a distributed library or object server system is that all of the distributed components must be at the same Content Manager OnDemand code level.

1.3.2 Content Manager OnDemand server components

A Content Manager OnDemand server environment contains several components:

- ▶ A *request manager* provides client, network, and operating system services, security, and accounting. The request manager is on the library server.
- ▶ A *database manager* maintains the index data for the reports that you store on the system. The database manager is a relational database management product, such as IBM DB2®. The database manager is on the library server.
- ▶ Database *control information* is information about the users, groups, application groups, applications, folders, cabinets, storage sets, and printers that you define on the system. The control information determines who can access the system, the folders that a user can open, and the application group data that a user can query and retrieve. The database is on the library server.
- ▶ A *cache storage manager* maintains documents in cache storage. If the archive storage server is accessed through the network, cache storage can be used for high-speed access to the most frequently used documents.
- ▶ An *Archive Storage Manager (ASM)* is an optional part of the system. The ASM is for the long-term storage of one or more copies of documents on archive media, such as slower disk or tape storage libraries.
- ▶ If your Content Management OnDemand System is installed on an MP platform and you need to download documents from a z/OS system, you can use a *download facility* to automatically transfer spooled files to the MP server. As a preferred practice, use Download for IBM z/OS, which is a licensed feature of IBM Print Services Facility™ (PSF) for z/OS. Download for IBM z/OS provides the automatic, high-speed download of Job Entry Subsystem (JES) spooled files from an z/OS system to Content Manager OnDemand servers. The download facility is not applicable to the IBM i server.
- ▶ *Data indexing and conversion programs* can create index data, collect required resources, and optionally convert Line Data reports to AFP data. Content Manager OnDemand provides several indexing programs:
 - The *Advanced Function Presentation (AFP) Conversion and Indexing Facility (ACIF)* can be used to index IBM z/OS Line Data, ASCII data, and AFP files, collect resources that are necessary to view the reports, and convert Line Data files to AFP data.
 - The *IBM OS/390 Indexer* is a high-performance indexer that can be used to index various data types and is available on both IBM z/OS and IBM AIX®.
 - The *IBM OS/400 Indexer* can be used to index various data types. It is the most common Content Manager OnDemand indexer for IBM i spooled files.
 - The *Content Manager OnDemand PDF Indexer* can be used to create index data for Adobe Portable Document File (PDF) files.
 - The *Content Manager OnDemand Generic Index File Format* can be used to provide index data for almost any other type of data, such as HTML documents, word-processing documents, and Tagged Image File Format (TIFF) files.

- The *XML Indexer* allows the rapid increase in XML archiving mandates that are based on ISO 20022 standards with XML (including SEPA in Europe). The XML Indexer is optimized for high-volume batch archiving of XML, batch PDF, AFP, Line Data, and check images.
 - The *Full Text Indexer* provides the capability to index the full text of a document (or report). You can search through an indexed document.
- ▶ *Data loading programs* can be set up to automatically store report data into application groups and update the database. The data loading programs can run on any Content Manager OnDemand server.
- ▶ *Report Distribution Facility* provides an easy way to automatically group reports and portions of reports and distribute the reports to multiple users. Distributions can be printed, created as an output file, or emailed as an attachment.
- ▶ Both the archived reports and their resources are stored in the Content Manager OnDemand Archive. The Content Manager OnDemand system manages the stored data throughout its lifetime. It provides authorized users rapid access to the data and allows the data to be converted into different formats for display or print.
- ▶ A *server print* facility allows users to reprint a large volume of documents at high speed. Print servers, such as Infoprint (on AIX), can be started to manage the server print devices. These print servers are not part of Content Manager OnDemand and must be purchased separately.
- ▶ Content Manager OnDemand *management programs* maintain the Content Manager OnDemand database and documents in cache storage.
- ▶ A *system logging* facility provides administrators with tools to monitor server activity and respond to specific events as they occur. The interface to the system logging facility is through the system log folder and the system log user exit.

Optimize Your IBM CMOD Return On Investment

End-to-end services from IBM's most experienced CMOD partner

Are you getting the maximum return on your IBM® Content Manager OnDemand (CMOD) investment? Is CMOD optimized to meet current and future requirements?

enChoice will ensure your system operates at peak performance. **Contact us** to learn why companies including Arizona Public Service, US Foods and Winnebago choose enChoice as their preferred partner for:

- Installations/deployments
- Migrations/upgrades
- Support/managed services
- Remote system administration
- Mentoring and V9.X education

FREE DOWNLOAD:

Top 10 Tips for Getting the Most Out of Your CMOD System

[Click here to read it now »](#)

enchoice.com | cmodinfo@enchoice.com

Premier Business Partner

enChoice®

THE ABOVE IS A PAID PROMOTION. IT DOES NOT CONSTITUTE AN ENDORSEMENT OF ANY OF THE ABOVE COMPANY'S PRODUCTS, SERVICES OR WEBSITES BY IBM. NOR DOES IT REFLECT THE OPINION OF IBM, IBM MANAGEMENT, SHAREHOLDERS OR OFFICERS. IBM DISCLAIMS ANY AND ALL WARRANTIES FOR GOODS OR SERVICES RECEIVED THROUGH OR PROMOTED BY THE ABOVE COMPANY.

THIS PAGE INTENTIONALLY LEFT BLANK



Setting up a Content Manager OnDemand instance

This chapter provides guidelines for implementing Content Manager OnDemand as a single instance.

In this chapter, we cover the following topics:

- ▶ Introduction
- ▶ Architecture and platform
- ▶ Implementing a Content Manager OnDemand instance on a multiplatform UNIX environment
- ▶ Implementing a Content Manager OnDemand instance on IBM i
- ▶ Implementing a Content Manager OnDemand instance on z/OS

2.1 Introduction

A *Content Manager OnDemand instance* is a logical server environment. The base system components are a library server and one or more object servers. Optional components include one or more archive managers and one or more Full Text servers.

2.2 Architecture and platform

Before you begin your installation and configuration, it is important to understand the general architecture of the Content Manager OnDemand server to help you determine the type of configuration that meets your business requirements. As illustrated in Figure 2-1, from an architectural perspective, the base Content Manager OnDemand server consists of two components: a library server and one or more object servers. The library server contains the database system tables and the application group data tables. The object server contains the stored reports and documents.

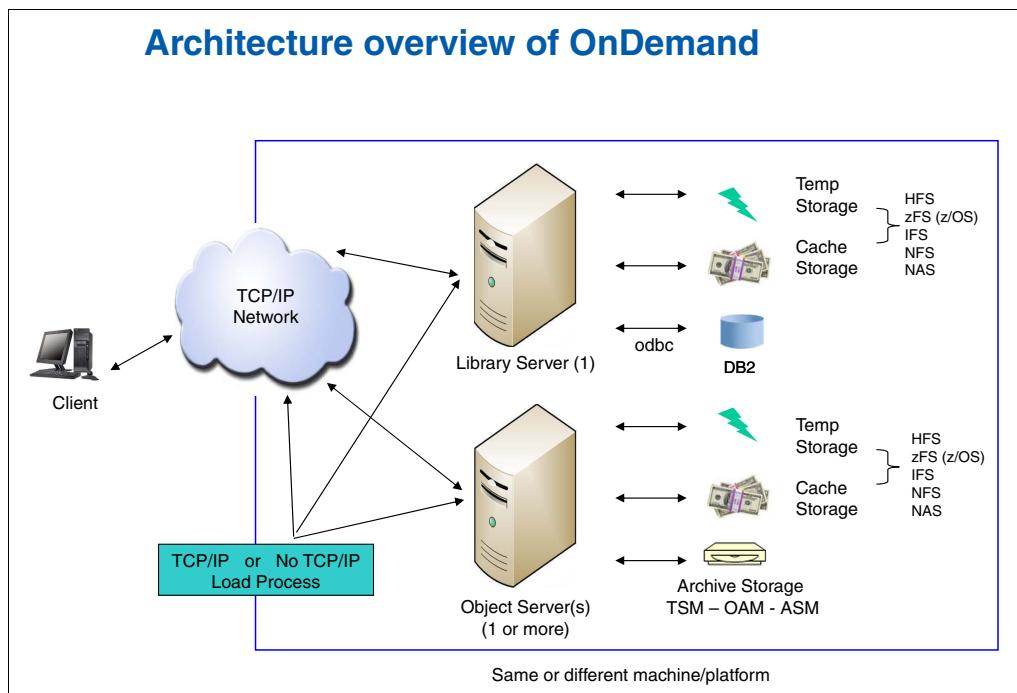


Figure 2-1 Architectural view of Content Manager OnDemand

Data is loaded and retrieved from the Content Manager OnDemand server through a network connection (TCP/IP). The advantages of this design are listed:

- ▶ The instance components can be physically distributed across the network.
- ▶ System users (client systems) can be anywhere on the network. By using the Internet, the clients can be anywhere in the world.
- ▶ The “load process” can run either on the library server, the object server, or on any other system (containing the appropriate software) that is attached by network to both the library and object server.
- ▶ This design also allows the library server and object servers to be placed on the same system (or logical partition (LPAR)) or on two (or more) systems (LPARs).

Even though the various components can be distributed across the network, the Content Manager OnDemand Instance appears as a “single system” to both the administrator and the users.

2.2.1 Configuration consideration

The basic Content Manager OnDemand configuration is to install the library server and object server on the same machine. This design is the default and most common configuration. However, under certain conditions, it might be beneficial to distribute the library server, the object server, and the load process to different systems.

Consider the following information to help you decide whether to run one or more of the load processes on one or more separate systems:

- ▶ Reducing system resource (processor or memory) consumption competition on the library and object server system.
- ▶ Reducing the network traffic time by running the load process on the system on which the load data is created.
- ▶ Better performance for a certain index or load process to run on another system.
- ▶ More convenient to operate and manage the data because the data is in a smaller set.

The considerations for separate object servers are listed:

- ▶ Distributing the workload among multiple systems.
- ▶ Distributing the data storage among multiple systems.
- ▶ Storing the data closer to where it will be accessed from. For example, your main operations (library server and object server) are in the US but many of your users are in China and France. You can install an object server in France to keep French data and another object server in China to store Chinese data. The original object server can remain in the US where the US data is kept.

The considerations for distributed library server, object servers, and load processes are listed:

- ▶ Using different operating systems. For example, the library server might be on a z/OS system while an object server can be on an AIX system and another object server can be on a Linux system.
- ▶ All of the system components must be at the same release level of Content Manager OnDemand, for example, Content Manager OnDemand 9.5.

2.2.2 Library server and object server functions

Functionality is distributed between the library server and object servers in the following manner:

- ▶ Library server:
 - Manages access to the administration definitions
 - Provides data integrity
 - Maintains data archive index information, configuration, and user account information
 - Controls access to data archives on object servers
 - Directs query, retrieve, and print requests from the clients
 - Routes store, retrieve, and delete requests from the clients

- Manages the (optional) Report Distribution System
- Manages the “interface” to the (optional) Full Text Index system
- Performs user authentication through internal security or external security System Authorization Facility (SAF) calls
- Performs logging
- ▶ Object server:
 - Provides the repository for Content Manager OnDemand data archives
 - Stores archive storage policy information
 - Manages the retention of Content Manager OnDemand data archives
 - Controls the transition of Content Manager OnDemand archives
 - Manages the expiration of Content Manager OnDemand archives

2.2.3 Choosing a platform

A Content Manager OnDemand server can run on many different operating systems and hardware environments. It can be set up to run on a workstation and can scale up to an IBM z™ Systems complex. The following factors need to be part of the decision to implement Content Manager OnDemand on one platform versus another platform:

- ▶ Existing hardware platforms
- ▶ Future hardware requirements (standardization, consolidation, and others)
- ▶ Existing personnel and skill set
- ▶ Current workload (number of users, quantity of data, and others)
- ▶ Future workload requirements (number of users, quantity of data, and others)
- ▶ Interfacing with other systems (software and data)
- ▶ Vendor’s ability to support the environment (hardware, software, and users over any geographic extent)

Default installation directory paths: The default installation directory path names changed for Content Manager OnDemand for Multiplatforms Server. The default installation paths for Content Manager OnDemand 9.5 are listed:

- ▶ /opt/IBM/ondemand/V9.5 for AIX and Sun (HP is no longer a supported platform in V9.5.)
- ▶ /opt/ibm/ondemand/V9.5 for Linux and Linux on IBM z Systems™
- ▶ C:\Program Files\IBM\OnDemand\V9.5 for Microsoft Windows
- ▶ /usr/lpp/ars/V9R5M0 for z/OS
- ▶ /QIBM/ProdData/OnDemand for IBM i

You can install Content Manager OnDemand to the default path or specify a different path. Because this book describes all Content Manager OnDemand platforms, you see various interchangeable references to these paths. Ensure that you check your own installation for the path name that is implemented and interpret the paths that are identified in the manual.

Starting with version 9.5, the installation can be performed by a non-root user on AIX. When installed as a non-root user, the installation path is fixed under the home directory of the user:

- ▶ \$HOME.\$HOME/IBM/ondemand/V9.5 for AIX and Sun
- ▶ \$HOME/ibm/ondemand/V9.5 for Linux and Linux on IBM z Systems

2.3 Implementing a Content Manager OnDemand instance on a multiplatform UNIX environment

In this section, we describe how to set up a single instance in a Content Manager OnDemand for a multiplatform UNIX environment. Always refer to the product documentation of your release for the specific steps to follow.

2.3.1 Defining a single instance

By default, the initial instance on any library server is named archive. Creating a single instance can be summarized by the following steps:

1. Creating a user
2. Creating a DB2 instance
3. Installing IBM Global Security Kit
4. Setting up Secure Sockets Layer (SSL)
5. Storing user IDs and passwords in a stash file
6. Installing and configuring Tivoli Storage Manager
7. Configuring the instance
8. Creating a Content Manager OnDemand database
9. Initializing the system log and system load facility

Creating a user

New installations (instances) of Content Manager OnDemand can be configured to run under a user other than the root user. If you plan to run an instance under a user other than root, complete the following steps:

1. Create the user for the Content Manager OnDemand instance owner that is a member of the database owners group.
2. Give the user administrator authority to the database.
3. Set permissions for the cache storage file systems.
4. Set permissions for the Content Manager OnDemand configuration and script files.
5. Give the instance owner permission to write to the system console.
6. Specify the instance owner in the ARS.INI file.

If you plan to run a distributed library and object server system, with one or more object servers on different workstations or nodes than the library server, you must also configure Content Manager OnDemand on the object servers.

To configure Content Manager OnDemand on the object servers, complete the following steps:

1. Create a group and user for the Content Manager OnDemand instance owner.
2. Give ownership of the cache storage file systems that are listed in the ARS.CACHE file to the group and user for the Content Manager OnDemand instance owner.

3. Give permission to read the following files to the Content Manager OnDemand instance owner:
 - ARS.CACHE
 - ARS.CFG
 - ARS.INI
 - ARS.DBFS
4. Give permission to write to the console to the Content Manager OnDemand instance owner.

Creating a DB2 instance

When you create a DB2 instance, you must install DB2 on the server. Complete the following steps:

1. Install IBM DB2 Universal Database™ Enterprise Edition.
2. Select **Typical** as the installation type to install all of the DB2 components that are required to support Content Manager OnDemand.
3. Create the DB2 instance for Content Manager OnDemand when you install DB2. Use the following values:
 - Instance Owner: archive.
 - Group Name: gname. The group must have SYSADM authority.
 - Home Directory: /home/archive.
 - Auto start DB2 instance at boot time: no.
 - Create a sample database for DB2 instance: no.

Installing IBM Global Security Kit

When you install the IBM Global Security Kit (GSKit), you can complete the task by using one of the following methods:

- ▶ SMIT GUI
- ▶ **installp** command

Before you perform the installation, complete the following steps:

1. Content Manager OnDemand is 64-bit application. Validate whether the Content Manager OnDemand Web Enablement Kit (ODWEK) installation is 32 bit or 64 bit.
2. Extract the correct GSKit media based on the version that is needed:
 - For the 32-bit version, run the following commands:


```
zcat gskcrypt32-8.0.14.45.aix.ppc.tar.Z | tar -xvf -
zcat gskssl32-8.0.14.45.aix.ppc.tar.Z | tar -xvf -
```
 - For the 64-bit version, run the following commands:


```
zcat gskcrypt64-8.0.13.4.aix.ppc.tar.Z | tar -xvf -
zcat gskssl64-8.0.13.4.aix.ppc.tar.Z | tar -xvf -
```

Setting up Secure Sockets Layer

Setting up Secure Sockets Layer (SSL) is optional. If you decide to use SSL, complete the following steps:

1. Create the key database and store it in the config subdirectory of the Content Manager OnDemand server installation directory:

```
/opt/IBM/ondemand/V9.5/config
```

To create the key database, run a command that is similar to the following command:

```
gsk8capicmd_64 -keydb -create -db "ondemand.kdb" -pw "myKeyDBpasswd" -stash  
-populate
```

2. Create a digital certificate.
3. Configure the Content Manager OnDemand for AIX server. Add the following lines to the ARS.INI file:

```
SSL_PORT=port_number  
SSL_KEYRING_FILE=/opt/IBM/ondemand/V9.5/config/ondemand.kdb  
SSL_KEYRING_STASH=/opt/IBM/ondemand/V9.5/config/ondemand.sth  
SSL_KEYRING_LABEL=IBM Content Manager OnDemand  
SSL_CLNT_USE_SSL=0
```

4. Restart the Content Manager OnDemand server.

Storing user IDs and passwords in a stash file

You can store user IDs and passwords in *stash* files (encrypted files). By storing passwords in a stash file, you can improve security because you do not need to specify the password on the command line, where the password might be visible to others.

To store the user IDs and passwords in a stash file, complete the following steps:

1. Create a stash file by running the **arsstash** command. The command prompts you for the password. For a description of the syntax of the **arsstash** command and an example of the command, see “Syntax of the ARSSTASH command” in the *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.
2. Save the stash file in a directory with limited access to that file. You can set this access by using file permissions.

When you configure the Content Manager OnDemand instance, you modify the ARS.INI file to include the **SRVR_OD_STASH** parameter and specify the directory that you specified in step 2:

```
SRVR_OD_STASH=/opt/IBM/ondemand/V9.5/config/ars.stash
```

Installing and configuring Tivoli Storage Manager

Before you begin, familiarize yourself with configuring and managing server storage, as described in the *Tivoli Storage Manager for AIX Administrator’s Guide*, GC32-0768. In addition, the *IBM Tivoli Storage Manager for AIX Administrator’s Reference*, SC32-0123, provides information about all of the commands that are used in this section. This book is your primary reference when you work with IBM Tivoli Storage Manager. If you encounter problems while you configure Tivoli Storage Manager or if the examples in this section do not provide the information that you need to define your server storage devices, policies, and operations, see these Tivoli Storage Manager publications.

To set up Tivoli Storage Manager for Content Manager OnDemand on an AIX workstation, complete the following steps:

1. Define the Tivoli Storage Manager server options.
2. Define the Tivoli Storage Manager client system options.
3. Define the Tivoli Storage Manager client options.
4. Register the Tivoli Storage Manager licenses.
5. Register the Tivoli Storage Manager administrators.
6. Define other Tivoli Storage Manager server options.
7. Start, halt, and restart the Tivoli Storage Manager server.
8. Increase the Tivoli Storage Manager database and recovery log sizes.
9. Define a storage library.
10. Define the policy domains.
11. Register the client nodes.
12. Prepare the storage pool volumes.
13. Optional: Configure Tivoli Storage Manager to maintain DB2 archived log files and backup image files.
14. Define a backup device for the Tivoli Storage Manager database.
15. Back up the Tivoli Storage Manager database and critical files.

Configuring the instance

Four configuration files must be updated or created. The files are installed with Content Manager OnDemand at installation time. For the AIX platform, the files are in the /opt/IBM/ondemand/<version>/config directory. The configuration files are listed:

- ▶ ARS.INI
- ▶ ARS.CFG
- ▶ ARS.CACHE
- ▶ ARS.DBFS

ARS.INI

The ARS.INI file contains a section for each instance; each section begins with a header. It is created at installation time, and by default, is configured with information for the archive instance.

Figure 2-2 on page 23 shows a sample ARS.INI file. In this scenario, OnDemand95 is the header line definition.

```
[@SRV @_ondmd950]
HOST=myServer.boulder.ibm.com
PROTOCOL=2
PORT=1450
SRVR_INSTANCE=ondmd950
SRVR_INSTANCE_OWNER=ondmd950
SRVR_OD_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.cfg
SRVR_DB_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.dbfs
SRVR_SM_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.cache
SSL_KEYRING_FILE=/opt/IBM/ondemand/V9.5/config/ondemand.kdb
SSL_KEYRING_STASH=/opt/IBM/ondemand/V9.5/config/ondemand.sth
SSL_KEYRING_LABEL=IBM Content Manager OnDemand
SSL_CLNT_USE_SSL=0
```

Figure 2-2 ARS.INI file sample

When you add an instance to an ARS.INI file, each instance must have a unique name. When you have more than one instance, you must identify the instance name when you are running Content Manager OnDemand programs (such as the **ARSLOAD**, **ARSDB**, and **ARSSOCKD** programs).

Figure 2-3 shows an example of an additional instance in the ARS.INI file.

```
[@SRV @_ondmd951]
HOST=myServer.boulder.ibm.com
PROTOCOL=2
PORT=1451
SRVR_INSTANCE=ondmd951
SRVR_INSTANCE_OWNER=ondmd951
SRVR_OD_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.v9501.cfg
SRVR_DB_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.v9501.dbfs
SRVR_SM_CFG=/opt/IBM/ondemand/V9.5/config/db2/ars.v9501.cache
SSL_KEYRING_FILE=/opt/IBM/ondemand/V9.5/config/ondmd951.kdb
SSL_KEYRING_STASH=/opt/IBM/ondemand/V9.5/config/ondmd951.sth
SSL_KEYRING_LABEL=IBM Content Manager OnDemand
SSL_CLNT_USE_SSL=0
```

Figure 2-3 Additional instance

Note: You must ensure that the ARS.INI file is consistent across *all servers* that are part of the Content Manager OnDemand system. If you update the ARS.INI file on the library server, you must update the ARS.INI files on the object server or servers if they are not on the same machines.

ARS.CFG

When an instance is started, Content Manager OnDemand reads the ARS.INI file to determine the location of the server configuration file. Each instance must have its own ARS.CFG file that is determined by the **SRVR_OD_CFG** parameter in ARS.INI.

When you configure the parameters, the default values are sufficient for most installations. However, you might need to adjust and tune the values to meet the requirements of your environment.

To update the ARS.CFG file, complete the following steps:

1. Log in to the server as the root user.
2. Change to the /opt/IBM/ondemand/V9.5/config directory.
3. Make a backup copy of the file that is provided by IBM.
4. Edit the ARS.CFG file.

Note for distributed library and object servers: Several parameters in the ARS.CFG file are not used on object servers. For example, an object server does not use the license parameters, server print parameters, and database parameters. For more information, see the following sections.

For distributed library and object servers:

1. Configure one copy of the ARS.CFG file on each workstation that is part of the Content Manager OnDemand system.
2. Set the **ARS_SRVR** parameter to the TCP/IP host name alias, fully qualified host name, or IP address of the library server.
3. Set the **ARS_LOCAL_SRVR** parameter to the TCP/IP host name alias, fully qualified host name, or IP address of the object server.
4. Save the file.
5. Set file permissions on the file to control access. Allow the Content Manager OnDemand instance to have read or write permissions.

Figure 2-4 on page 25 and Figure 2-5 on page 26 show sample ARS.CFG file content, which is split into two parts for easier reference.

```

# NOTE: See documentation for configuring these parameters.
#####
# OnDemand Parameters #
#####
# Number of client licenses (Library Server Only)
# - This should be set to however many licenses are purchased
ARS_NUM_LICENSE=100
# Language that is used to create the database (Library Server Only)
# - This should be set during installation and should never be changed
#
ARS_ORIGINAL_CODEPAGE=1208
#
ARS_SRVR=
ARS_LOCAL_SRVR=
# The number of Database SubServers to handle connections
#           to the database
ARS_NUM_DBSRVR=20
ARS_TMP=/arstmp/logs/db2/cmod95
ARS_PRINT_PATH=/arstmp/logs/db2/cmod95
#####
# Database Parameters #
#####
# Database for OnDemand to use (Library Server Only)
ARS_DB_ENGINE=DB2
# Used for arstblsp command and reloading migrated tables (Library Server
Only)
# 0 (import) 1 (load w/Tivoli Storage Manager - DB2 only) 2 (load w/DISK -
DB2 only, using ARS_TMP)
ARS_DB_IMPORT=0

```

Figure 2-4 ARS.CFG sample (part 1)

```

#####
# DB2 Parameters (Library Server Only) #
#####
DB2INSTANCE=archive
# These parameters are used only during database creation
ARS_DB2_DATABASE_PATH=/arsdb/cmod95
ARS_DB2_PRIMARY_LOGPATH=/arsdb_primarylog/cmod95
ARS_DB2_ARCHIVE_LOGPATH=/arsdb_archivelog/cmod95
ARS_DB2_LOGFILE_SIZE=1000
ARS_DB2_LOG_NUMBER=40
# DB2/Tivoli Storage Manager Parameters
ARS_DB2_TSM_CONFIG=/usr/tivoli/tsm/client/ba/bin/dsm.opt.db2
#####
# Storage Manager Parameters (Library/Object Server) #
#####
# Storage Manager for OnDemand to use
ARS_STORAGE_MANAGER=TSM
#ARS_STORAGE_MANAGER=CACHE_ONLY
#####
# Tivoli Storage Manager Parameters (Object Server Only) #
#####
DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
DSMI_LOG=/arstmp/logs/db2/cmod95
ARS_TRACE_SETTINGS=/opt/IBM/ondemand/V9.5/config/db2/cmod95.trace.settings
ARS_SUPPORT_FULL_TEXT_INDEX=1

```

Figure 2-5 ARS.CFG sample (part 2)

ARS.CACHE

Content Manager OnDemand supports cache storage for the temporary storage and high-speed retrieval of reports that are stored on the system. Each Content Manager OnDemand instance can have its own cache storage to allow a complete differentiation between the instances. The ARS.CACHE file identifies the file systems on the object server that can be used by Content Manager OnDemand for the cache storage.

Alternatively, the Content Manager OnDemand instance can share cache storage because Content Manager OnDemand separates the cache directories by first placing the instance name at the cache directory that is defined. For the archive instance, however, the cache directory is directly below the defined file system name. For the rest of the instance, the cache directories are separated by the instance name. The **SRVR_SM_CFG** parameter in the ARS.INI file identifies the cache file systems that are used by the instance. This file can contain one or more file systems.

Important: The first line in the ARS.CACHE file identifies the base cache storage file system where Content Manager OnDemand stores the control information. After you define this value, you cannot add or remove it from Content Manager OnDemand or change it in any way.

The cache file system must be owned by the Content Manager OnDemand instance owner. The permissions on these file systems are important. On AIX servers, the cache file system must be owned by the root user and the system group. On Linux, HP-UX, and Sun Solaris, these file systems must be owned by the root user and the root group. You must ensure that no other permissions are set. On AIX, the file system permissions need to be similar to the following example:

```
drwx----- 3 root system 512 Sep 22 13:08 /arscache/cache1
```

ARS.DBFS

The ARS.DBFS file location is identified in the ARS.INI file. It is read when the instance is started. The ARS.DBFS file contains the file names in which Content Manager OnDemand can store table spaces, and it determines the type of table space that Content Manager OnDemand can create. Storing application group index data in a table space is optional, but recommended. We also recommend that these file systems contain only Content Manager OnDemand data and that each instance on the server has its own file systems on which to store data. In general, the more table space file systems that you define, the better the system performance is. When you use more than one table space file system, each table space file system needs the same allocated disk space.

When you use DB2 as the database, Content Manager OnDemand supports the use of storage management subsystem (SMS) and automatic table space. See the DB2 documentation for the use of automatic table spaces. The use of the SMS table space allows the operating system to increase the size of the table space, as required, during a load process.

When you create an instance that uses table space, you must create an ARS.DBFS file. We created the ars.dbfs file in our scenario. Each line in this file must contain the name of the file system and the type of table space to be stored. The naming convention of these files is shown:

```
/filesystem SMS
```

The name of the file system must identify the table spaces that can be created in the file system. For example, the following line identifies the SMS table space file system:

```
/arsdb/db1/SMS SMS
```

These file systems must be owned by the database instance owner and the group. In our scenario, the file system is owned by db2iadm1 and belongs to the sysadm1 group. See the following example for the correct permissions:

```
drwxrws--- 4 archive db2iadm1 512 May 17 12:58 /arsdb/db1/SMS
```

We include the SMS in the file system name to indicate the type of data that is stored.

Creating a Content Manager OnDemand database

After the database instance is created, and all of the Content Manager OnDemand directories are set up with the appropriate permissions, it is time to create the Content Manager OnDemand database. Verify that the group to which the database instance owner (ondtest) belongs has write access to the database directory names that are specified in the ARS.CFG file.

The **arsdb** command performs the following actions:

- ▶ Updates the database configuration.
- ▶ Verifies the directories for the primary and the archived log files.
- ▶ Creates a link to the database user exit program.
- ▶ Creates a backup of the database.

- ▶ Builds the Content Manager OnDemand system tables and indexes.
- ▶ Binds the database to Content Manager OnDemand.

Sign on to the user account that you assigned as the owner of the Content Manager OnDemand instance (in the ARS.INI file). Run **arsdb** with the following options:

```
/opt/IBM/ondemand/V9.5/bin/arsdb -I ondmd950 -cv
```

In our scenario, -I ondmd950 is the Content Manager OnDemand instance.

After this command completes, you can log in to DB2 and connect to the new instance. List all of the tables by running the following command:

```
db2 list tables for all
```

Initializing the system log and system load facility

After you create the database, you can initialize the system log by running the following command:

```
/opt/IBM/ondemand/V9.5/bin/arssyscr -I ondmd950 -l
```

-I ondmd950 is the new Content Manager OnDemand instance.

Content Manager OnDemand can track loading activity with the system load logging facility. Content Manager OnDemand stores these load messages in the system load log. You can initialize the system load log by running the following command:

```
/opt/IBM/ondemand/V9.5/bin/arssyscr -I ondmd950 -a
```

Again, -I ondmd950 is the new Content Manager OnDemand instance.

The **arssyscr** program creates the application groups, applications, and folders that are required by the system logging facility.

Note: The **arsdb** and **arssyscr** commands are in /opt/IBM/ondemand/V9.5/bin for AIX, HP-UX, and Sun Solaris, and in /opt/ibm/ondemand/V9.5/bin for Linux.

2.3.2 Starting and connecting to the new instance

After the instance is created, you can start the new instance and connect to it.

Starting and stopping arssockd

To start the instance manually, run the following command and include the instance name after the **arssockd** command:

```
/opt/IBM/ondemand/V9.5/bin/arssockd -I ondmd950 -Sv
```

Run the **ps** command to verify that the instance is started:

```
ps -ef | grep ars
```

If more than one instance is running, you see more than one **arssockd** process in the display. The instance other than the default instance archive has a **-instancename** after **arssockd** for identification:

```
OnDemand95 65864128      1  0   Jun 11      - 0:00 arssockd-ondmd950:
```

Ensure that when you stop the instance, you stop the correct instance. You might stop the instance by running `kill` on the process identifier (PID) of the accepting process or by running the following command:

```
arssockd -I ondmd950 -T
```

Connecting to an instance

To connect to an instance, the client must log on to the correct library server. Add a server in the administrative or user client by identifying the name of the library server and the port number to use. The port number that you specify must be the same port number that you specified in the ARS.INI file.

Running commands

In general, the `-h` or `-I` parameters are used to determine the name of the Content Manager OnDemand instance to process. You must specify the parameter and the instance name if any of the following items are true:

- ▶ The name of the default instance is not ARCHIVE.
- ▶ You are running more than one instance on the same system and you want to process an instance other than the default instance.
- ▶ You are running the program on a system other than the system where the library server is running.

The programs locate the specified instance name in the ARS.INI file to determine the TCP/IP address, host name alias, or fully qualified host name of the system on which the library server is running and other information about the instance. The **ARSADM**, **ARSADMIN**, **ARSDOC**, and **ARSLOAD** programs support the `-h` parameter. The **ARSDB**, **ARSLOAD**, **ARSMaint**, and **ARSTBLSP** programs support the `-I` parameter.

For the **ARSLOAD** program, if both the `-h` and `-I` parameters are specified, the value of the *last* parameter that is specified is used, for example:

```
arsload -g applicationgroup -u userid -p password -I ondmd950 test.data  
arsmaint -cmsv -I ondmd950
```

2.4 Implementing a Content Manager OnDemand instance on IBM i

Always refer to the product documentation of your release for the specific steps to follow. A Content Manager OnDemand instance is a logical server environment with its own library that contains a unique set of database files. An instance is defined in the ARS.INI file by naming the instance, which identifies the name of the library that is used by the instance. All of the database files that belong to an instance are created in only one Coded Character Set ID (CCSID).

You can run multiple instances on the same server, with each instance configured differently:

- ▶ For separate test and production environments
- ▶ For databases that use different CCSIDs

In addition, each Content Manager OnDemand for i instance must run in a single CCSID.

When you work with more than one instance, you must identify the instance name when you run Content Manager OnDemand commands, such as **ADDRPTOND** and **STRMONOND**. You can specify the instance name as a command parameter each time that you run a command. Or, you can set up a data area on your system that identifies the default instance name and then specify only the instance name as a command parameter when you must specify a name other than the default.

Each instance has different security from other instances on the same machine. You define users and groups to each instance and set application group and folder permissions for users in each instance. Each instance has its own system log.

2.4.1 Configuring the instance

To create your Content Manager OnDemand instances, complete the following steps:

1. Your user profile must have its locale set to the locale of the instance that you want to create.

Because the locale is set in the user profile, you might need to change your user profile, sign off, and sign on again before you create the instance. Use the Change User Profile (**CHGUSRPRF**) command to change your user profile, if necessary. Also, ensure that other language-related parameters in your user profile are set correctly. (The Change User Profile (**CHGUSRPRF**) command does not show the current locale setting; it shows *SAME. Run the Display User Profile (**DSPUSRPRF**) command to check the locale setting.)

The Locale Job Attributes (**SETJOBATR**) parameter in your user profile is used to determine the values that are obtained from the locale. For Content Manager OnDemand, at a minimum, you must use **SETJOBATR(*CCSID)**. For example, if you are in the US and are using the English language, you enter the following Change User Profile command (all as one command):

```
CHGUSRPRF USRPRF(user_profile_name) LANGID(ENU) CNTRYID(US) CCSID(37)
SETJOBATR(*CCSID *DATFMT *TIMSEP *DATSEP *DECfmt *SRTSEQ)
LOCALE('/QSYS.LIB/EN_US.LOCALE')
```

If you are in Spain and are using the Spanish language with euro support, you enter the following command (all as one command):

```
CHGUSRPRF USRPRF(user_profile_name) LANGID(ESP) CNTRYID(ES) CCSID(1145)
SETJOBATR(*CCSID *DATFMT *TIMSEP *DATSEP *DECfmt *SRTSEQ)
LOCALE('/QSYS.LIB/ES_ES.LOCALE')
```

For more information about locales, see Chapter 13, “Defining a locale”, in the *IBM Content Manager OnDemand for i - Common Server Planning and Installation Guide*, SC19-2790.

2. Choose a name for the instance or use the default instance name of QUSROND.

The instance name must be a valid library name for IBM i. The instance name must start with an alphabetic character or @ followed by any of these characters: 0 - 9, A - Z, @, #, or underscore (_). Ensure that no library, user profile, or authorization list by that name exists. The instance name must not start with the letter Q (except for QUSROND), and must not be named CONFIG, GUI, or WWW. This instance name is referred to as [instance] in the rest of these instructions.

3. Create the instance by using the Create Instance for Content Manager OnDemand (**CRTINSTOND**) command.

At a minimum, you must specify the name of the instance (which then uses system values and defaults for the additional parameters, such as *DFT for the **PORT** parameter, which uses port 1445). You can specify additional parameters to customize the instance to meet your requirements.

For example, you can specify a three-character language identifier (by using the **LANGID** parameter), which must match one of the language identifiers that are listed in Chapter 13, “Defining a locale”, in the *IBM Content Manager OnDemand i - Planning and Installation Guide*, SC19-2790. If you specify the **LOCALE** parameter, the one that you specify must be included in the list of valid locales that are listed in Chapter 13, “Defining a locale”, in the *IBM Content Manager OnDemand i - Planning and Installation Guide*, SC19-2790.

If the instance is in a user auxiliary storage pool (ASP), the user ASP number (2 - 32) must be specified for the **ASP** parameter and *ASP must be specified for the **ASPDEV** parameter. If the instance is in an independent auxiliary storage pool (IASP), *ASPDEV must be specified for the **ASP** parameter and the IASP name (such as IASP2) must be specified for the **ASPDEV** parameter.

For example, the Create instance for Content Manager OnDemand command **CRTINSTOND INSTANCE(ONDTEST) LANGID(ENU) LOCALE('/QSYS.LIB/EN_US.LOCALE')** creates an instance that is called ONDTEST with a server language of US English that uses TCP/IP port 1445.

The **CRTINSTOND** command performs the following actions:

- Creates the **/CONFIG** directory under **/QIBM/UserData/OnDemand** and the default and model files under **/QIBM/UserData/OnDemand** (if they do not exist).
- Appends the model **ARS.INI** file (in **/QIBM/ProdData/OnDemand/config**) to the actual **ARS.INI** file (in **/QIBM/UserData/OnDemand/config**) and uses the name of the instance wherever it finds [instance] in the model file.
- Creates the instance directory **/QIBM/UserData/OnDemand/[instance]**. If the instance is in an Independent ASP, the instance directory path is preceded by the Independent ASP name. For example, if the Independent ASP name is IASP, the instance directory is created in **/IASP/QIBM/UserData/OnDemand**.
- Creates the **ARS.CFG**, **ARS.CACHE**, and **ARS.DBFS** files in **/QIBM/UserData/OnDemand/[instance]** and uses the name of the instance wherever it finds [instance] and the language identifier wherever it finds [language] in the model file. (The model files for these three files are in **/QIBM/ProdData/OnDemand/config**.) If the instance is in an Independent ASP, the instance directory path is preceded by the Independent ASP name. For example, if the Independent ASP name is IASP, the **ARS.CFG**, **ARS.CACHE**, and **ARS.DBFS** files are created in **/IASP/QIBM/UserData/OnDemand/[instance]**.
- Creates the library and database tables for the instance. If the instance is in an IASP, you must set the ASP Group before you can work with files in that library. Run the Set ASP Group (**SETASPGRP**) command to set the ASP Group.
- Creates the directories that are needed for the instance as specified in the **ARS.CFG** and **ARS.CACHE** files.
- Creates a user profile with the same name as the instance, and adds that user to the instance as a Content Manager OnDemand system administrator.
- Adds the user **QONDADM** to the instance as a Content Manager OnDemand system administrator.

- Creates an authorization list with the same name as the instance.
- If the instance is in an Independent ASP, a record is added to the file QARLCASP in library QUSRRDARS.

2.4.2 Changing an instance configuration

To change an instance configuration, you must change the configuration parameters in the ARS.INI and ARS.CFG files.

Configuration parameters in the ARS.INI file

You might need to change several of the configuration parameters from the values that you specified when you ran the **CRTINSTOND** command before you use this instance for the first time. If so, you must edit the ARS.INI file in the /QIBM/UserData/OnDemand/config directory by running the following command:

```
EDTF '/QIBM/UserData/OnDemand/CONFIG/ARS.INI'
```

You can change the values that are listed below. The instance definition starts with the line **[@SRV0_[instance]]**, where **[instance]** is the name of the instance. For example, the instance ONDTEST starts with the line **[@SRV0_ONDTEST]**.

The following lines might need to be reviewed:

- ▶ **PORT=0:** The port to which the server listens to receive requests from a Content Manager OnDemand client. The value of 0 means to use the default port of 1445. Only one server can be listening to a particular port at a specific time. To run multiple instances concurrently, you must specify an unused port on your system. You can run the Work with TCP/IP Network Sts (**WRKTCPSTS**) **OPTION(*CNN)** command to see the ports that are in use on your system.
- ▶ **SRVR_FLAGS_SECURITY_EXIT=1:** This line specifies that you want to use IBM i user IDs and passwords as the Content Manager OnDemand user IDs and passwords. This method is the default value, which makes it simpler for your users because they do not have to maintain multiple passwords. If your Content Manager OnDemand users do not need to have IBM i user IDs, specify a value of 0 for this parameter. When you specify 0, your Content Manager OnDemand passwords have no relationship to IBM i passwords.

However, if a Content Manager OnDemand user ID and an IBM i user profile match, certain Content Manager OnDemand commands and APIs use the IBM i user profile as the Content Manager OnDemand user ID, even if you chose not to relate the two user IDs. This situation can permit IBM i users to perform Content Manager OnDemand functions that you did not intend for them to perform. Therefore, a Content Manager OnDemand user ID must not match an IBM i user profile name unless the two IDs are used by the same individual.

If you change the **SRVR_FLAGS_SECURITY_EXIT** value, review the Content Manager OnDemand System Parameters values (defined by the Content Manager OnDemand Administrator Client) for the instance that you changed. For more information, see “OnDemand userid relationship to IBM i user profiles” in the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.

- ▶ **HOST=LOCALHOST:** If you are enabling IPv6 on your IBM i system and need certain Content Manager OnDemand instances to use IPv4 addressing and others to use IPv6 addressing, you might need to change HOST=LOCALHOST to HOST=IPV6-LOCALHOST within the ARS.INI stanza for each instance that you want to use IPv6 addresses. You might want certain instances to run with IPv6 and others with IPv4. This mixed environment is fully supported. Also, during the transition from IPv4 to IPv6, Content Manager OnDemand clients that use IPv4 addresses can connect to the server simultaneously with clients that use IPv6 addresses.

Configuration parameters in the ARS.CFG file

You might need to change several of the ARS.CFG configuration parameters from the default values before you use this instance. To do so, edit the ARS.CFG file in the /QIBM/UserData/OnDemand/*instancename* directory (where *instancename* is the name of the instance that you want to review). For example, you might use the following Edit File command:

```
EDTF '/QIBM/UserData/OnDemand/MYINSTANCE/ARS.CFG'
```

If the instance is created in an IASP that is named IASP2, use the following command:

```
EDTF '/IASP2/QIBM/UserData/OnDemand/MYINSTANCE/ARS.CFG'
```

You can change these values:

- ▶ **ARS_LANGUAGE=ENU:** Specifies the language in which this instance runs. The 'ENU' value indicates the usage of the English language. Valid languages are listed in the "Locales" section in the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.
- ▶ **ARS_MSGS_LANGUAGE=ENU:** Specifies the language that is used for server messages. The 'ENU' value indicates the usage of the English language. Valid languages are listed in the "Locales" section in the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.
- ▶ **ARS_AUTOSTART_INSTANCE=1:** Specifies whether to automatically start the server for this instance when you use the Start TCP/IP Server (**STRTCPSSR**) command. Set this value to 1 to automatically start this instance's server; set this value to 0 if you do not want to automatically start this instance's server. For more information about controlling the servers that start automatically, see the "Starting and stopping servers" section in the *IBM Content Manager OnDemand for i - Common Server Planning and Installation Guide*, SC19-2790.

Do not modify any of the other values in these instance definition files without first consulting with Content Manager OnDemand Support.

You must end and restart the instance server after you make any changes.

2.4.3 Starting and stopping servers

You must start a server for an instance before clients can connect to it.

Starting the servers

Servers are started by running **STRTCPSVR *ONDMD**. The **INSTANCE** parameter of the **STRTCPSVR *ONDMD** command supports the special values of ***DFT**, ***ALL**, and ***AUTOSTART**, and the specification of the name of an instance. (An instance is set to autostart if the **ars.cfg** file for that instance contains **ARS_AUTOSTART_INSTANCE=1**.) The default value for the **INSTANCE** parameter is ***DFT**. You can also create a data area that is named **STRTCPSVR** to further control the behavior of the **STRTCPSVR** command. For more information about the data area, see the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.

Without the **STRTCPSVR** data area, the values of ***DFT** and ***AUTOSTART** work identically. All instances that are set to autostart are started. Use the special value ***ALL** to start all of the instances that are configured on the system. You can also specify the name of a single instance to start, for example:

```
STRTCPSVR SERVER(*ONDMD) INSTANCE(ONDTEST)
```

With the data area, the value of ***DFT** starts only the instance that is named in the data area. The data area must be named **STRTCPSVR** and in library **QUSRDRARS**. The data area must be of the type character with a length of 10. To create the data area, run the following command (all as one command):

```
CRTDTAARA DTAARA(QUSRDRARS/STRTCPSVR) TYPE(*CHAR) LEN(10) VALUE(QUSROND)  
TEXT('Autostart instance name for STRTCPSVR *ONDMD *DFT')
```

QUSROND is the name of the instance to start.

The special values ***ALL** and ***AUTOSTART** work the same with the data area as without the data area.

To determine the instances that are started when **STRTCPSVR SERVER(*ONDMD) INSTANCE(*AUTOSTART)** is run, you can look for the **ARS_AUTOSTART_INSTANCE=1** in the **ARS.CFG** file. However, an easier way is available so that you do not need to check the **ARS.CFG** file for every instance.

Run **grep** in Qshell to search the contents of all of the **ARS.CFG** files for the string **ARS_AUTOSTART_INSTANCE=1**, for example:

```
$  
grep -n 'ARS_AUTOSTART_INSTANCE=1' /qibm/userdata/ondemand/*/ars.cfg  
/qibm/userdata/ondemand/ONDDEMO/ars.cfg:53:ARS_AUTOSTART_INSTANCE=1  
/qibm/userdata/ondemand/ONDDEU/ars.cfg:53:ARS_AUTOSTART_INSTANCE=1  
/qibm/userdata/ondemand/ONDENU/ars.cfg:53:ARS_AUTOSTART_INSTANCE=1  
/qibm/userdata/ondemand/QUSROND/ars.cfg:53:ARS_AUTOSTART_INSTANCE=1  
$
```

From the last four detail lines, which are the output of the **grep** command, you can determine that instances **ONDDEMO**, **ONDDEU**, **ONDENU**, and **QUSROND** are started when the **STRTCPSVR SERVER(*ONDMD) INSTANCE(*AUTOSTART)** command is run.

Table 2-1 on page 35 summarizes the behavior of the **STRTCPSVR** command with and without the **STRTCPSVR** data area.

Table 2-1 Behavior of the STRTCPSVR command with or without the STRCPSVR data

Running STRTCPSVR start	*DFT	*ALL	*AUTOSTART	Named instance
Without the data area	All instances set to autostart	All instances that are configured on the system	All instances set to autostart	The named instance
With the data area	Only the instance that is named in the data area	All instances that are configured on the system	All instances set to autostart	The named instance

Stopping the servers

Servers are stopped by running ENDTCPYSR *ONDMD. The instance parameter of the STRTCPSVR *ONDMD command supports the special values of *DFT and *ALL, and the specification of the name of an instance. The default value for the INSTANCE parameter is *DFT. You also can create a data area that is named STRTCPSVR to further control the behavior of the ENDTCPYSR command. Create the data area as described in “Starting the servers” on page 34. For more information about the data area, see the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792. Even though the data area is named STRTCPSVR, it controls both the STRTCPSVR and ENDTCPYSR commands by design so that *DFT starts and ends the same instance.

Without the STRTCPSVR data area, the values of *DFT and *ALL work identically. All instances that are active are ended. You can also specify the name of a single instance to end, for example:

```
ENDTCPYSR SERVER(*ONDMD) INSTANCE(ONDTEST)
```

With the data area, the value of *DFT ends only the instance that is named in the data area. The data area must be named STRTCPSVR and in library QUSRRDARS.

Table 2-2 summarizes the behavior of the ENDCPSVR command with and without the data area.

Table 2-2 Behavior of the ENDCPSVR command with or without the data area

Running ENDCPSVR ends	*DFT	*ALL	Named instance
Without the data area	All active instances	All active instances	The named instance
With the data area	Only the instance that is named in the data area	All active instances	The named instance

Server work management

Server jobs are started by using a job description with the name of the instance, which must be in the instance library. If a job description with that name is not found in the instance library, job description QOND400 in library QRDARS is used (and can be changed if necessary).

The job description controls the following attributes of the server job:

- ▶ JOBQ
- ▶ JOBPTY
- ▶ OUTPTY
- ▶ PRTDEV
- ▶ OUTQ

- ▶ INLLIBL
- ▶ LOG
- ▶ LOGCLPGM
- ▶ INQMSGRPY
- ▶ JOBMSGQMX
- ▶ JOBMSGQFL

For example, if you want to change the job queue that instance TEST uses, you create a job description that is called TEST in the TEST library that specifies the job queue that you want to use.

To change the run priority of the server jobs, you must add a routing entry to the subsystem. The server job is always submitted with routing data QRLMSERVER.

To change the run priority of all server jobs for all instances to 40, add the following routing entry to subsystem QSYSWRK. (You must choose a sequence number (SEQNBR) that is not already in use.)

```
ADDRTGE SBSD(QSYSWRK) SEQNBR(1841) CMPVAL(QRLMSERVER) PGM(QSYS/QCMD)
CLS(QSYS/QSYSCLS40)
```

After you make this change, you must stop and restart all of your servers.

Automatically starting instances

To enable an instance to start automatically each time that the system restarts, you must add one of the commands that are described in 2.4.3, “Starting and stopping servers” on page 33 to your **QSTRUP** program. You can also add the commands to a job scheduler.

2.5 Implementing a Content Manager OnDemand instance on z/OS

Instances on z/OS do not differ greatly from instances on Multiplatforms. The concept is the same. In this section, we explain how to set up a new instance and provide background information about the UNIX System Services implementation. Always refer to the product documentation of your release for the specific steps to follow.

Before you set up your z/OS instance of Content Manager OnDemand, understand the different types of configurations and the components that make up the Content Manager OnDemand instance. A source for determining this information is the *IBM Content Manager OnDemand for z/OS - Introduction and Planning Guide*, SC19-365.

Instances are logical implementations for the separation of administration functions, users, and data on the same server. Instances have the same physical access to the program libraries, but they have different databases with a separate system log and separate file systems. Instances are typically used to separate different customers on one z/OS server to separate the test and production environments, or to use different code pages on different databases.

A Content Manager OnDemand instance on a z/OS server is a separately started task (ARSSOCKx) that uses different databases, users, and application groups. Every user on the instance must be defined for the instance. Every instance has its own security if internal security is used. If an external security exit is used, it is common over all of the instances.

Figure 2-6 on page 37 shows an overview of the single instance on z/OS.

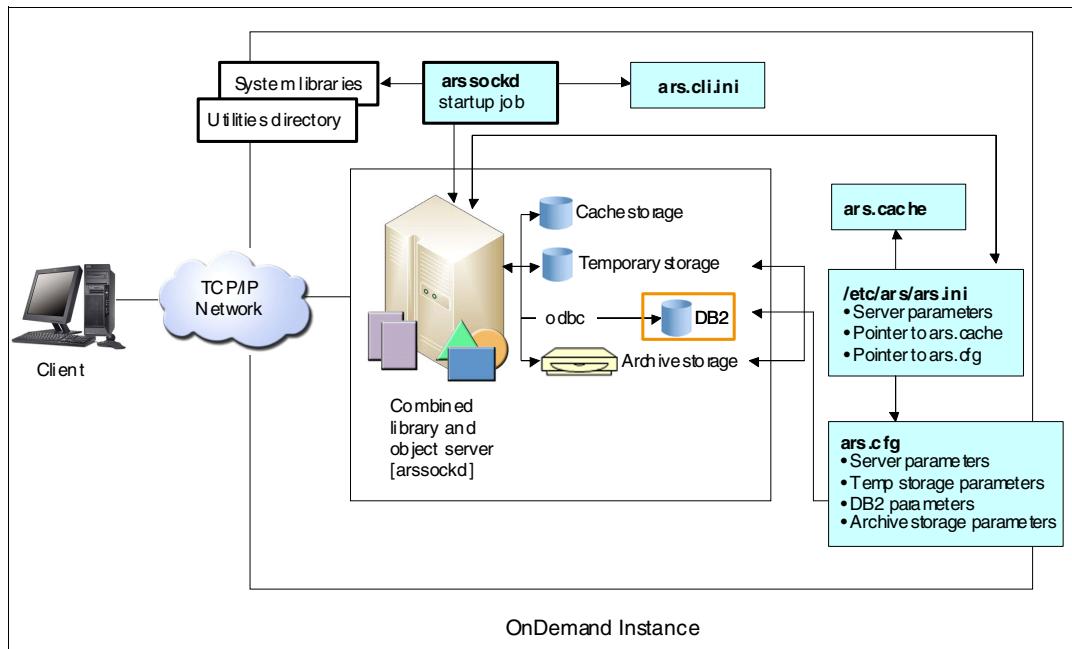


Figure 2-6 Single instance overview on z/OS

2.5.1 Installation overview

The path for the Content Manager OnDemand system is /usr/lpp/ars/V9R5M0 (on z/OS) and /opt/IBM/ondemand/V9.5 (on UNIX). From the ars directory, several directories contain the Content Manager OnDemand files and executable files, such as programs and procedures. The directories are created during the installation when you run the ARSMKDIR REXX routine from the installation library, ARS.V9R5M0.SARINST. The /usr/lpp/ars/V9R5M0 directory contains the subdirectories that are listed in Table 2-3.

Table 2-3 Subdirectories of /usr/lpp/ars

Directory	Content
bin	All executable files, such as arsdb for creating the database
config	All configuration datasets, such as ARS.INI
locale	All subdirectories for globalization
MidServer	All configuration files for Structured APIs
samples	All sample files for updating
www	All subdirectories for ODWEK

Important: All path parameters and commands are *case-sensitive*.

Sometimes when you choose a directory, such as /usr/lpp/ars/V9R5M0/bin, you see a different path when you run **pwd** because a symbolic link is set. A *symbolic link* is a file that contains the path name for another file or directory. Only the original path name is the real name. An *external link* is a type of symbolic link; it links to an object outside of the hierarchical file system (HFS). Typically, it contains the name of an IBM MVST™ dataset.

2.5.2 Creating an instance on z/OS

In this section, we explain how to create an instance on the z/OS system. To do so, complete the following steps:

1. Copy the control files.
2. Verify the ARS.INI file.
3. Verify the ARS.CFG file.
4. Modify the ARS.CACHE file.
5. Verify the CLI.INI file.
6. Modify the **ARSSOCKD** procedure.
7. Modify the **ARSLOAD** procedure.

You can mount the Content Manager OnDemand installation directory at any mount point other than /usr/lpp/ars/V9R5M0. You can run at different service levels with this flexibility. For example, a symmetric multiprocessor (SMP) might be used to install into SERVICE/usr/lpp/ars/V9R5M0. SERVICE/usr/lpp/ars/V9R5M0 might be copied into /usr/lpp/ars/V9R5M0/maint for testing. When testing is complete, /usr/lpp/ars/V9R5M0/maint might be copied into /usr/lpp/ars/V9R5M0 for production.

Copying the control files

To copy the control files, complete the following steps:

1. Create a directory (/etc/ars) for maintaining the updated configuration files.
2. Create a symbolic link from the installed directory /usr/lpp/ars/V9R5M0/config to the /etc/ars directory, for example, ln -s /etc/ars /usr/lpp/ars/V9R5M0/config.
3. Set the appropriate access mode of 755.

ARS.INI

The ARS.INI file contains a section for each instance; each section begins with a header. It is created at installation time and, by default, it is configured with information for the archive instance. In this scenario, ARC95037 is the header line definition.

Figure 2-7 shows the content of a sample ARS.INI file.

```
[@SRV@_ARC95037]
HOST=MyHost
PROTOCOL=2
PORT=1937
SRVR_INSTANCE=ARSDB937
SRVR_INSTANCE_OWNER=ARSUS937
SRVR_OD_CFG=/usr/lpp/ars/V9R5M0/config/ars937.cfg
SRVR_SM_CFG=/usr/lpp/ars/V9R5M0/config/ars937.cache
SSL_KEYRING_STASH=/usr/lpp/ars/V9R5M0/config/ars937.stash
SRVR_FLAGS_SECURITY_EXIT=0
SRVR_FLAGS_FOLDER_APPLGRP_EXIT=0
SRVR_FLAGS_DOCUMENT_EXIT=0
SRVR_FLAGS_SQL_QUERY_EXIT=0
SRVR_FLAGS_FORCE_SECURITY=0
```

Figure 2-7 ARS.INI file sample

ARS.CFG

When an instance is started, Content Manager OnDemand reads the ARS.INI file to determine where the server configuration file is. Each instance must have its own ARS.CFG file that is determined by the **SRVR_OD_CFG** parameter in ARS.INI.

For a distributed library server or object server, you must configure a copy of the ARS.CFG file on each distributed server.

When you configure the parameters, the default values are sufficient for most installations. However, you might need to adjust and tune them to meet the requirements of your environment.

Figure 2-8 shows the content of a sample ARS.CFG file.

```
ARS_ORIGINAL_CODEPAGE=
ARS_LOCAL_SRVR=
ARS_NUM_DBSRVR=20
ARS_NUM_LICENSE=1
ARS_NUM_OAMSRVR=20
ARS_OAM_DB2SSID=DSNA
ARS_OAM_PLAN=CBRIDBS
ARS_PRINT_PATH=/ars/tmp
ARS_SRVR=
DB_ENGINE=DB2
ARS_LDAP_ALLOW_ANONYMOUS=TRUE /* Allow anonymous bind connections */
ARS_LDAP_BASE_DN=foo /* Specifies 'foo' as base distinguished name */
*/
ARS_LDAP_BIND_ATTRIBUTE=bar /* Specifies 'bar' as bound attribute */
ARS_LDAP_BIND_MESSAGES_FILE='$ONDEMAND/LDAP/msg_string.txt' /*
Specifies location of LDAP message string file */
ARS_LDAP_MAPPED_ATTRIBUTE=foonly /* Specifies attribute 'foonly',
returned to \
OnDemand as user ID */
ARS_LDAP_PORT=389 /* Specifies port on which LDAP listens */
ARS_LDAP_SERVER=127.0.0.1 /* Specifies IP address of LDAP server */
```

Figure 2-8 ARS.CFG file sample

ARS.CACHE

The ARS.CACHE file identifies the file systems on the server that can be used by Content Manager OnDemand for the cache storage. The file system is in the HFS or z/OS file system (zFS) storage.

Figure 2-9 shows an example of an ARS.CACHE file that specifies two cache storage file systems.

```
/ars1
/ars2
```

Figure 2-9 ARS.CACHE file sample

Important: The first line in the ARS.CACHE file identifies the base cache storage file system where Content Manager OnDemand stores the control information. After you define this value, you cannot add or remove it from Content Manager OnDemand or change it in any way.

The SRVR_SM_CFG parameter in the ARS.INI file identifies the cache file systems that are used by the instance. This file can contain one or more file systems.

Configuring LDAP (optional)

Optionally, you can configure Lightweight Directory Access Protocol (LDAP) to control the logon access to Content Manager OnDemand. LDAP is needed for multiplatform implementation and IBM i.

After you configure LDAP, you can then configure Content Manager OnDemand to access the LDAP server by modifying the following configuration files:

- ▶ ARS.CFG
- ▶ ARSLDAP.INI

For more information, see Chapter 11, “Configure LDAP”, of the *IBM Content Manager OnDemand for z/OS Configuration Guide*, SC19-3363.

Verifying CLI.INI

The CLI.INI file contains the information that relates to the Open Database Connectivity (ODBC) connection that the ARSSOCKD program uses to connect to DB2. This information is referenced by the DSNA0INI DD statement in the PROC job control language (JCL) or as an HFS file.

Figure 2-10 displays an example of the CLI.INI.

```
[COMMON]
MVSDEFAULTSSID=DSNA
[DSNA]
PLANNAME=DSNACLI
```

Figure 2-10 CLI.INI file sample

The following information refers to Figure 2-10:

- ▶ MVSDEFAULTSSID=DSNA: Identifies, to ODBC, the DB2 subsystem name or group attachment name in a data sharing group.
- ▶ PLANNAME=DSNACLI: If the plan for ODBC is not the default plan DSNACLI, you must specify the plan with PLANNAME=plan.

Modifying the ARSSOCKD procedure

The ARSSOCKD started task is used to start the server. You must customize the procedure for your environment and copy it in the PROCLIB concatenation. A copy of the procedure is provided in the SARSINST library.

Modifying the ARSLOAD procedure

The ARSLOAD procedure is used to start the ARSLOAD started task that monitors the spool and loads into Content Manager OnDemand.

A sample ARSLOAD procedure is provided in the SARSINST library.

Creating the database for the new instance

The new instance uses its own set of tables. A new database must be created for this new instance. Perform this task by modifying the ARSDB2 member in the SARSINST library, which is used for the initial installation. Several modifications of this job are necessary:

- ▶ Change the SQLID to another user, who must have sysadm authority.
- ▶ Optional: Change the Create Storage Group Statement if you want a new storage group for the instance.
- ▶ Change the Create Database Statement.
- ▶ Run the job.

Creating the table space

The new instance uses its own tables, so table space is needed. You can accomplish this task by modifying the ARSTSPAC member in the SARSINST library, which is used for the initial installation. Make the following modifications:

- ▶ Change the SQLID to the same user that creates the database in ARSDB2.
- ▶ Change the **IN** parameter of the **CREATE TABLESPACE** statement to the database name that you previously created in ARSDB2.
- ▶ Change the **USING** parameter to the STOGROUP name that you previously used in ARSDB2.
- ▶ Set the appropriate values for the primary and secondary allocation.
- ▶ Run the job.

Creating tables for the new instance

After all of the DB2 objects are created and the configuration files are updated, the database for the instance (the system tables) must be created. Use the **arsdb** program to create the database for the instance.

Important: You must identify the instance name when you run the Content Manager OnDemand programs, such as the **arsdb**, **arsload**, and **arssockd** programs, and when you run the database commands.

To create the tables, complete the following steps:

1. Go into OMVS to set up the ODBC environment.
2. Switch to the superuser (SU).
3. Set the environment variables to access DB2 on z/OS by running the following command:

```
export DSNA0INI="/etc/ars/cli.ini"
```

The minimum parameters are the DB2 SSID and the interface (DSNCLI).
4. Run **SET** from the OMVS command line.
5. Move to the Content Manager OnDemand executable directory by running the following command:

```
cd /usr/lpp/ars/V9R5M0/bin
```

6. Run the **ARSDB** program. This program name is *case-sensitive*.

```
arsdb -I ARC95037 -c
```

7. The **ARSDB** program generates a series of messages. It acknowledges the successful creation of the tables when all of the tables are created without any error; otherwise, it creates error messages.

You might see a message that is similar to the following example:

```
arsdb: "Unable to determine the database engine"
```

This message might look like a DB2 error, but the **ARSDB** program cannot read the configuration file. Check the log for any IBM RACF® messages that are writing to or opening the file system.

Many installations run several DB2 systems on the z/OS logical partition (LPAR). Sometimes, this approach can lead to errors if the link list contains only the DSNLOAD and DSNEXIT library from a different DB2 subsystem. You can add your requested DB2 library by running the **export** command:

```
export STEPLIB="SYS1.DSNA.SDNSLOAD:SYS1.DSNA.SDSNLOAD2:SYS1.DSNA.SDSNEXIT"
```

This command sets the environment.

Tip: If you exit the shell, the setting is gone. You can add the **export** command to your OMVS login profile. Check your variables by running **SET**.

Initializing the system log, system load, and system migration

After you create the Content Manager OnDemand system tables, the system log must be initialized with the **ARSSYSCR** program for this new instance. To do so, complete the following steps:

1. Move to the Content Manager OnDemand executable directory by running the following command:

```
cd /opt/IBM/ondemand/V9.5/bin
```

2. Run the **ARSSYSCR** program for this instance and use the **-I** parameter:

```
arssyscr - I ARC95037 -l
```

Here, ARC95037 is the name of the instance.

Content Manager OnDemand supports the ability to track loading activity with the system load logging facility. Content Manager OnDemand stores these load messages in the system load log. You can initialize the system load log by completing the following steps:

1. Move to the Content Manager OnDemand executable directory by running the following command:

```
/opt/IBM/ondemand/V9.5/bin
```

2. Run the **ARSSYSCR** program for this instance and use the **-I** parameter:

```
arssyscr - I ARC95037 -a
```

Again, **-I ARC95037** is the new Content Manager OnDemand instance.

The system migration facility is required if you plan to migrate application group index data from the database to the archive. You initialize the system migration facility by completing the following steps:

1. Move to the Content Manager OnDemand executable directory by running the following command:

```
/opt/IBM/ondemand/V9.5/bin
```

2. Run the **ARSSYSCR** program for this instance and use the **-I** parameter:

```
arssyscr - I ARC95037 -m
```

Again, **- I ARC95037** is the new Content Manager OnDemand instance.

The **ARSSYSCR** program creates the application groups, applications, and folders that are required by the system logging, system load, and system migration facilities.

2.5.3 Starting and verifying the new instance

Now that the new instance is set up, you can start it and verify that it is installed correctly.

Starting the new instance

When everything is set up, you can start the new instance by customizing the sample procedure in the SARSINST library to conform to your environment.

Figure 2-11 shows an example of starting the new instance.

```
//ARS95037 PROC PARML=
//*
//** Library: USER.PRIVATE.PROCLIB(ARS95037)
//*
//ARS95037 EXEC PGM=ARSSOCKD,REGION=0M,TIME=NOLIMIT,
// PARM='/VERBOSE ARC95037'
//STEPLIB DD DISP=SHR,DSN=ARS.ARSV950.SARSLOAD
//          DD DISP=SHR,DSN=DSN.DB2V910.SDSNEXIT
//          DD DISP=SHR,DSN=DSN.DB2V910.SDSNLOAD
//          DD DISP=SHR,DSN=DSN.DB2V910.SDSNLOD2
//ARSBIN DD PATH='/usr/lpp/ars/V9R5M0/bin'
//DSNAOINI DD PATH='/etc/ars/cli937.ini'
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
```

Figure 2-11 Sample Content Manager OnDemand procedure

After this procedure is started, log on to the new instance by using the different port number and create users, application groups, applications, and storage sets with the normal procedures.

Running **arsload** to check the new instance and new file system

After all of the configuration work is complete and the application group, application, and folder are created, run **arsload** for installation verification. Figure 2-12 on page 44 shows the procedure that is used to load data to the new instance. If you see problems in loading the file (writing an object), check the user permissions.

```
//ARSLOAD PROC
//ARSLOAD EXEC PGM=ARSLOAD,REGION=0M,TIME=NOLIMIT,
// PARM=(’/-h ARC95037 -C Q’)
//STEPLIB DD DISP=SHR,DSN=ARSV950.AE.SARSLOAD
//          DD DISP=SHR,DSN=SYS1.DB1K.SDSNEXIT
//          DD DISP=SHR,DSN=SYS1.DB1K.SDSNLOAD
//          DD DISP=SHR,DSN=SYS1.DB1K.SDSNL0D2
// DD DISP=SHR,DSN=ACIF.V4R3M0.SAPKMOD1
//*****
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=121,BLKSIZE=6050
//SYSOUT DD SYSOUT=*
//*****
//** The following 2 DD statements should be uncommented and
//** customized if the PDF indexer is used.
//*****
//**ADBERES DD DSN=ADOBE.PDFLIB.RESOURCE.INDEX(ADBERES),DISP=SHR
//**ADBEFNT DD DSN=ADOBE.PDF405.PLUSP1C.ADOBEFNT.LST,DISP=SHR
```

Figure 2-12 ARSLOAD for new instance



Administration

An important aspect of a Content Manager OnDemand system is the effective design and implementation of a strategy for system administration from a report administration perspective and from a user authority and responsibility perspective. The focus of this strategy is to ensure that the system is planned to provide the greatest functionality and the best performance as the system matures.

In this chapter, we cover the following topics:

- ▶ Report administration
- ▶ User and group administration
- ▶ XML Batch Administration

3.1 Report administration

Report design and definition are key to a successful implementation of a Content Manager OnDemand system. Knowledge of the data that will be indexed, loaded, and retrieved, with knowledge of Content Manager OnDemand preferred practices, results in the most efficient and easy-to-use system possible. In this section, we consider the processes that are followed when you define a Content Manager OnDemand report. We present hints and tips to help in the design and implementation process.

The system components that are required for creating, retrieving, and viewing a Content Manager OnDemand report are a storage set, an application group, an application, and a folder. Optionally, cabinets might be used to organize and simplify folder access. These elements, in combination, allow the Content Manager OnDemand administrator to define and create a report definition that can then be used to index and load data into Content Manager OnDemand. Figure 3-1 illustrates the relationship of these elements in a typical Content Manager OnDemand system.

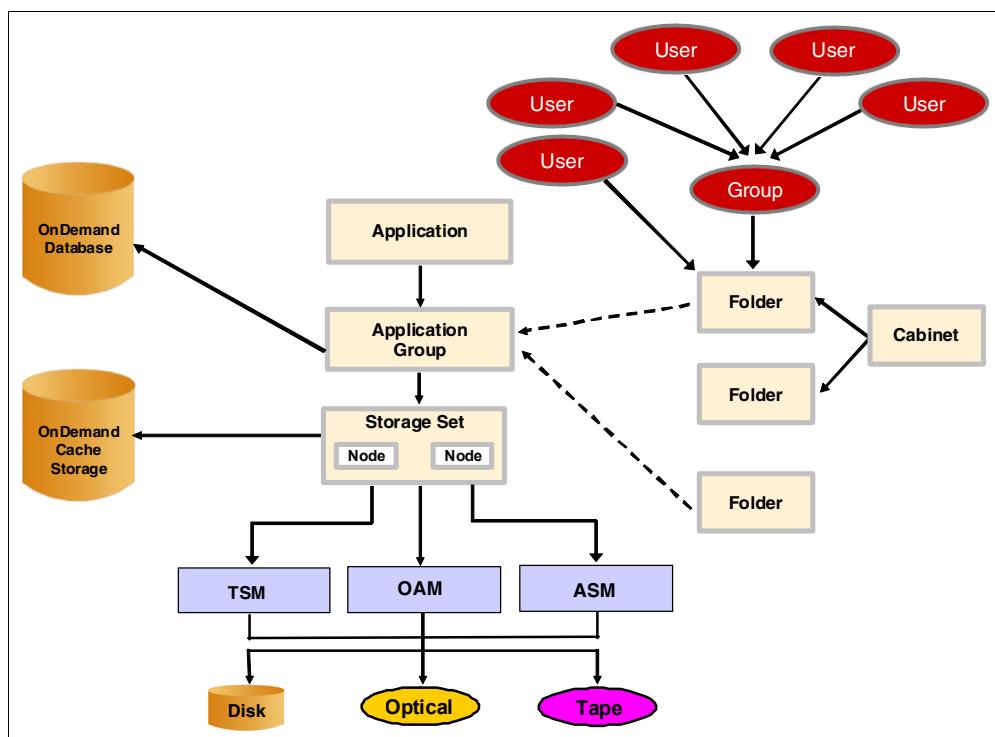


Figure 3-1 Content Manager OnDemand system components relationship

To help you better understand how to perform report administration, we use the example company that is mentioned in 1.2.1, “Background information of an example company” on page 6 with the Content Manager OnDemand Administrator Client running on Windows to create the required system components. We use the monthly credit card statements that are generated by AFInancial Co in our example. These statements are stored in a single application group in Content Manager OnDemand.

3.1.1 Storage sets

When you define a report, the first component to create is a storage set if one does not exist. A *storage set* is a named collection of primary storage nodes that support application groups with similar archive storage management requirements.

For example, a storage set can be used to maintain data from different application groups that must retain documents for the same length of time and require the data to be kept on the same type of media. Different storage sets can be created to handle different data retention requirements. One storage set can be set up to maintain data on cache-only storage, and another storage set can be set up to point to an archive storage to maintain data for three years on optical media. Business practices and legal requirements determine the storage management design that is required.

For most implementations of Content Manager OnDemand for i, a storage set is created by creating a migration policy.

Figure 3-2 shows the window where you add a storage set. When you define a storage set, you must specify the report load type. For our example, we create a storage set, StatementStorage. It stores credit card statements from the customers. The load type is Fixed.

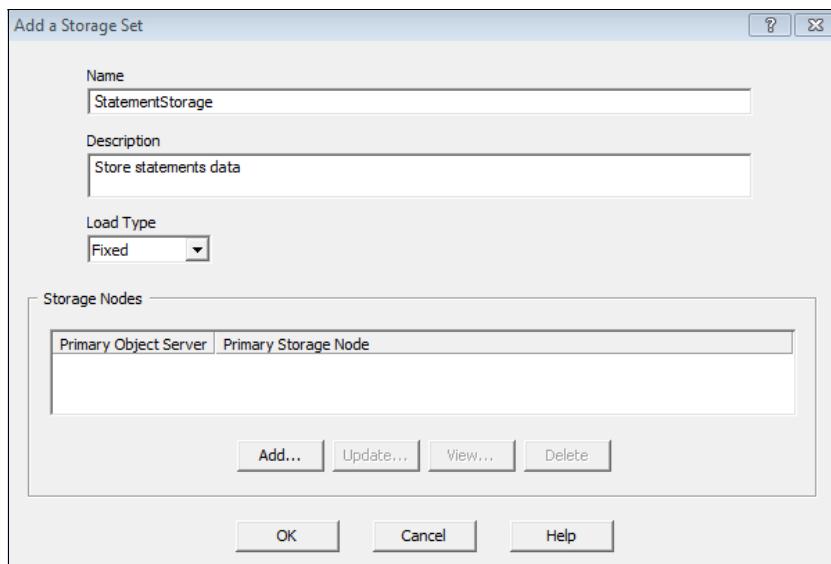


Figure 3-2 Creating a storage set

For a more in-depth look into storage management, see Chapter 5, “Storage management” on page 89.

3.1.2 Application groups

After you create a storage set, the next object to create is an application group. We create an application group that is called Credit Card Statements. An *application group* represents the data that you store in Content Manager OnDemand. It contains a collection of one or more applications that contain common indexing and storage management requirements. The application group contains the database information that is used to load, search for, and retrieve reports. The application group also defines the data that will be loaded into the database.

Figure 3-3 on page 48 shows the Add an Application Group window. For our example, we create an application group that is called Credit Card Statements. All applications that are related to credit card statements are under this application group.

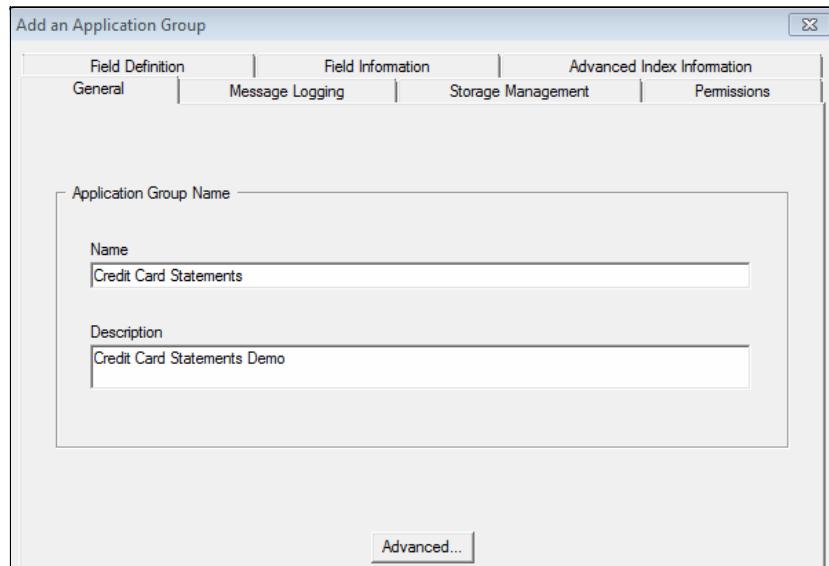


Figure 3-3 Creating an application group

When you define an application group, you also specify the database fields and storage management information. In the following sections, we look closer at different aspects of the application group definition that can contribute to a successful Content Manager OnDemand system implementation.

Database information

Click Advanced (shown in Figure 3-3) to specify advanced database information for the application group. Figure 3-4 shows the Database Information window.

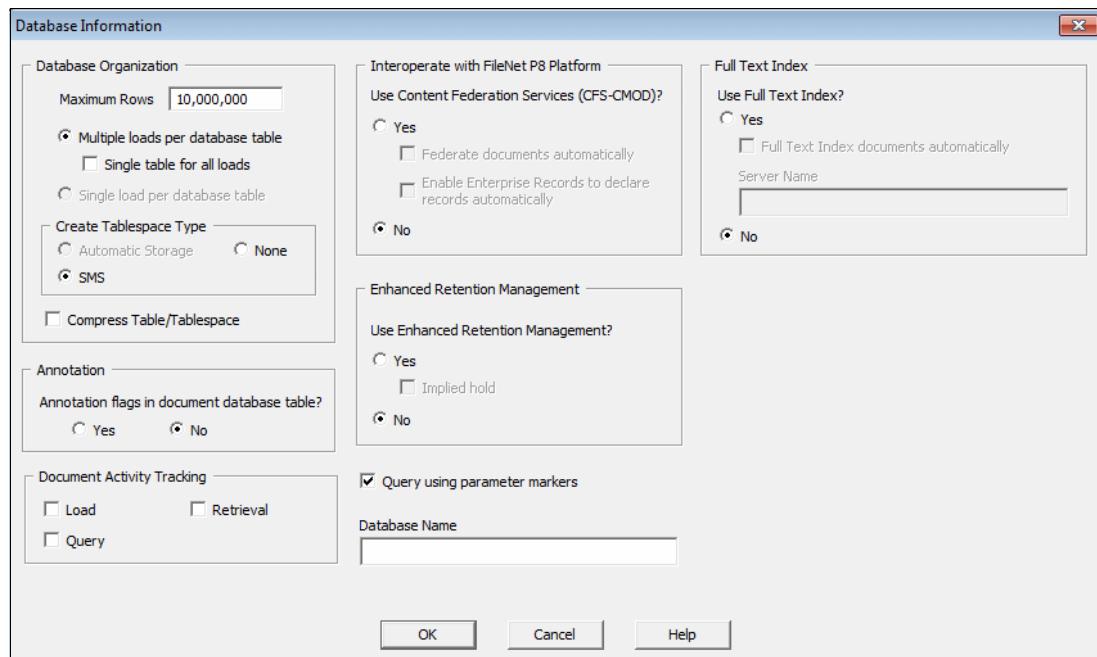


Figure 3-4 Database information on a z/OS system

When you define the database information, you must determine the number of rows (Maximum Rows) that are stored in each database table and the number of report loads that are included in each database table. These values are important to system performance and maintenance.

Maximum Rows value

Physically, an application group consists of one or more tables that contain the index data for the stored documents and reports. Each table is referred to as an *application group segment table*. The *Maximum Rows* value determines how many data rows are loaded into each of these “segment” tables. When the Maximum Rows value is reached, the open segment table is closed and a new segment table is created. We selected the default value of 10 million rows because in most cases, this value is sufficient for balancing the performance of data loads and queries.

Consider the following factors when you specify the Maximum Rows value:

- ▶ The number of rows that is specified needs to be large enough to handle the largest possible input report file.
- ▶ Decrease the value if the total number of indexes (documents) that is stored for the application group will never reach the 10 million rows value.
- ▶ Increase the Maximum Rows value if the 10 million rows value is too small compared to the number of indexes (documents) to be stored so that many application group segment tables are created and as a result, a single user query results in a search of multiple tables.

Storage Management

The Storage Management setting determines how long report data and indexes are kept in cache storage before they are expired. Options determine how soon data is migrated to archive storage after the report load completes.

For a detailed description of the storage management options, see Chapter 5, “Storage management” on page 89.

Field Information tab

The Field Information tab (Figure 3-5) is used to define the attributes of the database fields that make up the Content Manager OnDemand report index data. These attributes determine the characteristics of the index data and control many aspects of loading and processing data in the system. A database field must be added for each index value that is required by applications to be part of the application group.

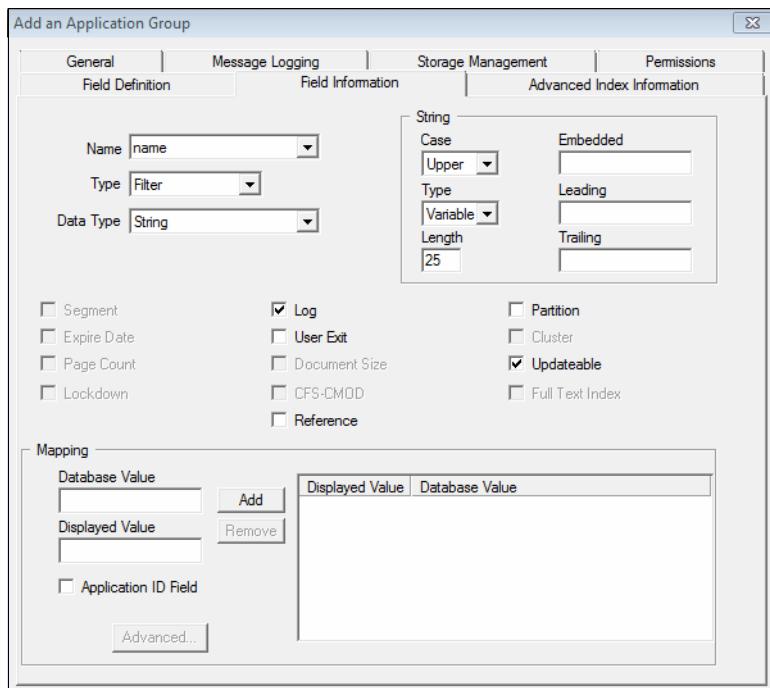


Figure 3-5 Application group's Field Information tab

Note: You can update the application group to add a database field.

When you define a field, you specify the field name, field type, data type, whether it is a segment field, the expiration date, and more. In the following sections, we take a closer look at field type and whether it is a segment or an application ID field.

Type

The Type attribute determines the manner in which the database field is used by Content Manager OnDemand. The main types of attributes are *index* and *filter*.

A field needs a type of *index* if it is used to uniquely identify a document or if it is frequently used when a user searches for documents in the application group. Designating a field as an index serves to enhance query performance but increases the required processing during loading and database maintenance. A separate index table is created and maintained for application group fields that are designated as indexes. These index tables are searched first when a *folder* query is run to quickly pinpoint the documents to include in the document "hit" list.

A field needs a type of *filter* if it does not uniquely identify a document. A filter is used with an index field during folder queries.

Important: Folder queries that use filter fields alone result in a sequential scan through database tables. An index field must always be included in folder queries. For more information about folders, see 3.1.4, “Folders” on page 54.

A thorough understanding of the way that users search for documents in the system is required before you determine the fields that will be indexes and the fields that will be filters. Only fields that will be heavily used when users are searching for and retrieving documents need a type of index. An index field will always be included in a folder query.

Note: Date fields are almost always defined as filters, not indexes.

Segment

Segment is the date or date and time field that is used to limit the number of tables that are searched during a folder query. By using a segment date to limit folder queries to a single table or a limited set of tables, performance is improved. The segment date is especially important for application groups that contain a large amount of data.

If the expiration type is segment, Content Manager OnDemand also uses the segment field to determine when to delete data from the application group. You might specify only one segment field for each application group.

Note: The date field that is used for the segment date must always have a type of *filter*. By default, an index is created for the segment date, and setting the segment date to a type of index creates unnecessary processing.

Application ID field

The application ID field is used to identify an application within an application group when you create an application group that contains more than one application. The database mapping fields are used to map the value to be stored in the database as the label that is displayed for folder queries and in the subsequent query hit list. A query can be made against a specific application in an application group or against all of the applications in an application group.

3.1.3 Applications

An application defines the data to index and load. An application associates the data with an application group and specifies the type of indexing process to perform on the data. It also defines any logical views to be put in place for users and determines any special print options to use with the data. In this section, we consider several of the load information attributes that are defined for an application.

Load Information tab

The Load Information tab specifies the processing and resource information that the Content Manager OnDemand loader uses to load the input data onto storage volumes and to load the associated index data into the database.

The File Format, Preprocessor Parameters, and Postprocessor Parameters (Figure 3-6) are defined as part of the load information:

- ▶ **File Format:** Provides settings that control how the Content Manager OnDemand system compresses and stores documents and resources
- ▶ **Preprocessor:** Specifies processing that is carried out on database fields before indexing data
- ▶ **Postprocessor:** Specifies a system command or exit program that runs against an index file before the index records are loaded into the database

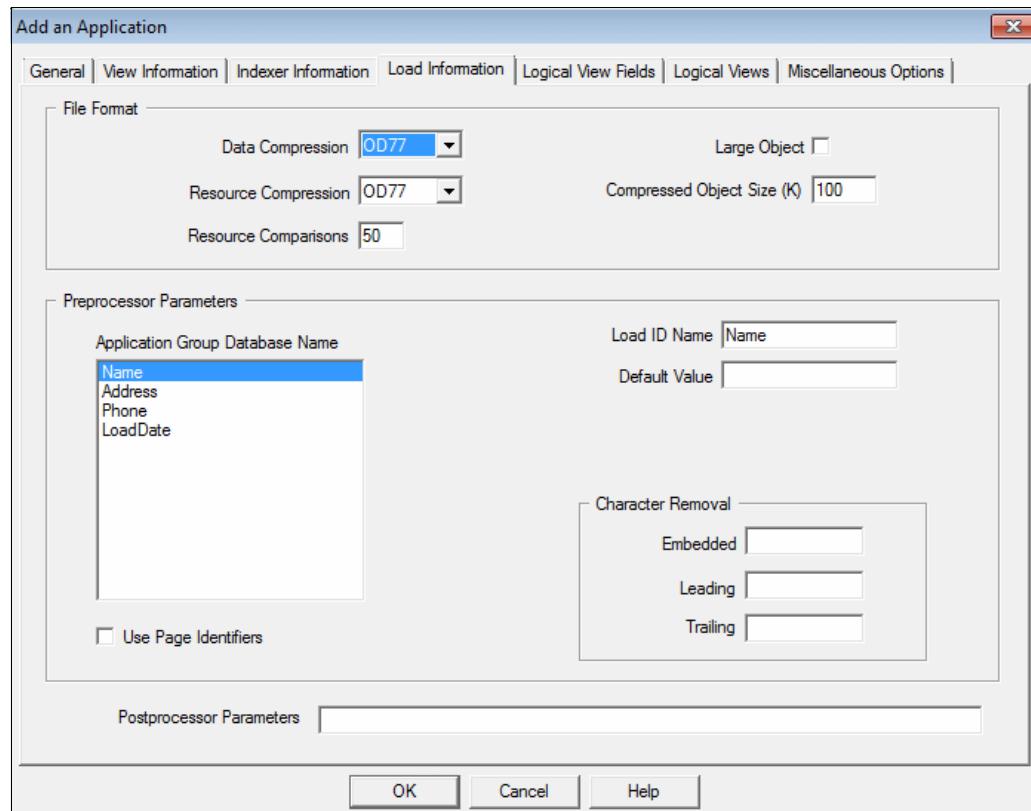


Figure 3-6 Application Load Information

Large object

In the File Format section, you can set support for large objects. Content Manager OnDemand large object support provides enhanced usability and better retrieval performance for reports that contain large documents.

For example, suppose that a report contains statements that typically exceed 1,000 pages. With large object support, the statements can be divided into parts of 100 pages. When a user views a statement, Content Manager OnDemand retrieves and uncompresses the first part of the statement. To view a specific page of a statement, the user can choose the Go To command in the viewer window and enter the page number. Content Manager OnDemand automatically retrieves and uncompresses the part of the statement that contains the requested page. When the user moves from page to page of a statement, Content Manager OnDemand automatically retrieves and uncompresses parts of the statement as needed.

When you use large object support, users experience consistent response time when they move from page to page of the document.

You must consider several factors when you use large object support:

- ▶ The report must be indexed with an indexing program that generates a large object by dividing large documents into smaller parts and defining the indexing information that is used to retrieve the documents.
- ▶ The amount of data per page and the number of pages that you divide documents into affects retrieval and viewing response time. The number of bytes per page typically dictates the number of pages that you can divide documents into. In general, the larger the page size in bytes, the smaller the number of pages that you can divide your documents into. For example, if the average page in the document contains 2.5 KB of data, choose 100 - 1000 pages per Large Object (LO) segment; if the average page in the document contains 50 KB of data, choose 1 - 100 pages per LO segment.
- ▶ The capacity of your network and the traffic in the network might determine the number of pages that you need to divide your documents into. Larger document sizes (large byte size even when compressed) require more network bandwidth (or more time if the bandwidth is not available) to transfer from a Content Manager OnDemand server to a client. The number of users that are concurrently accessing Content Manager OnDemand and the sizes of the documents that are being retrieved determine the overall load in the network.
- ▶ Response time requirements. The goal of Content Manager OnDemand large objects is to provide better performance and usability. Large object support clearly provides enhanced usability. However, you must implement large object support so that dividing your documents into parts provides better overall performance than other methods of segmenting the input data.

When you choose a large object, Content Manager OnDemand displays the Number of Pages field. Specify the number of pages that you want Content Manager OnDemand to divide documents into in the Number of Pages field.

To generate large objects, the indexer that is specified on the Indexing Information page must be AFP Conversion and Indexing Facility (ACIF), OS/390, or OS/400. When you select the Large Object check box, Content Manager OnDemand automatically adds the INDEXOBJ=ALL parameter to the indexing parameters (which causes the indexing program to generate the large object indexing information).

Exporting an application

It is not possible to export an application to application groups with different database fields or attributes. However, you can export applications to a different server if the application group on the target server is identical to the application group on the source server (the server on which the applications are defined).

Ensure that no existing application has the same application ID in the target application group. For more information, see the section “Adding items to a server” in the *IBM Content Manager OnDemand for Multiplatforms, V9.5, Administration Guide*, SC19-3352.

Selecting font by line data graphical indexer

The font that is used by the line data graphical indexer to display a document can be changed from within the line data graphical indexer at the Content Manager OnDemand Administrator Client.

Note: For the best indexing results, select a monospacing font with the line data graphical indexer.

If the font is changed by using the Administrator Client, the selected font is also used by the Windows client the next time that the Windows client is started and a line data document is viewed.

For more information, see Technote 1215957, which is available at the following web address:
<http://www.ibm.com/support/docview.wss?uid=swg21215957>

3.1.4 Folders

A *folder* is the interface that allows a user to search for reports and documents that are stored in the Content Manager OnDemand system. One or more application groups can be defined to a folder. The user enters index search criteria into the folder search fields. In the background, an SQL search is issued for each included application group. The results of the queries are accumulated, and a document hit list is constructed and returned to the user. The folder can be customized to provide the look and feel that is wanted for the users of the Content Manager OnDemand system. The Content Manager OnDemand administrator can also grant specific permissions for users and groups to use the folders.

Figure 3-7 shows the Add a Folder window.

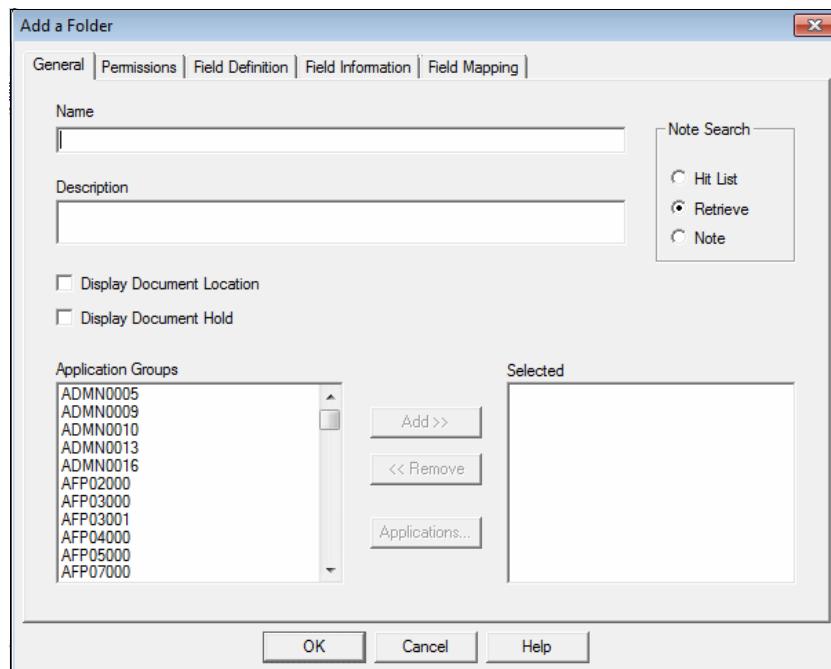


Figure 3-7 Folder general information

Display Document Location

The Display Document Location setting (Figure 3-7) determines whether the client shows the storage location of each document in the document list by placing an icon next to each entry. The possible locations are cache storage (on the library server or an object server) or archive storage.

Important: Use care when you enable this feature. The Display Document Location function can result in degraded search performance because the storage location information for every document that is returned must be retrieved from the Content Manager OnDemand object server.

Display Document Hold

The Display Document Hold setting (Figure 3-7 on page 54) determines whether the client shows a column that indicates whether a hold is placed on the document. For more information, see Chapter 16, “Enhanced Retention Management” on page 353.

Note Search

If the annotation parameter (annotation flags in the document database table) in the application group is set to “No”, the Note Search parameter (Figure 3-7 on page 54) determines when Content Manager OnDemand searches the database for annotations and notifies the user of the annotations. The following options are possible:

- ▶ Hit list: When a folder query is run, Content Manager OnDemand searches for annotations, and a note icon, which contains an annotation, is displayed next to each document in the resulting hit list. The hit list option has a direct performance impact on the generation of the document list.
- ▶ Retrieve: Content Manager OnDemand searches for annotations when the user selects a document for display. This option is the default and preferred option.
- ▶ Note: Content Manager OnDemand searches for annotations when the user selects the **note** command when the user views a displayed document.

As a preferred practice, set the annotation parameter in the application group advanced settings to “Yes”. In this case, an annotation flag is set in the database when a user adds an annotation to a document. When the document hit list is displayed, a note icon is displayed next to the documents for which an annotation exists.

Full Report Browse

In the Permissions tab of the folder definition window (Figure 3-8 on page 56), the Full Report Browse option allows a user of the Content Manager OnDemand Windows Client to select a document, retrieve that document, and view the entire report to which the document belongs.

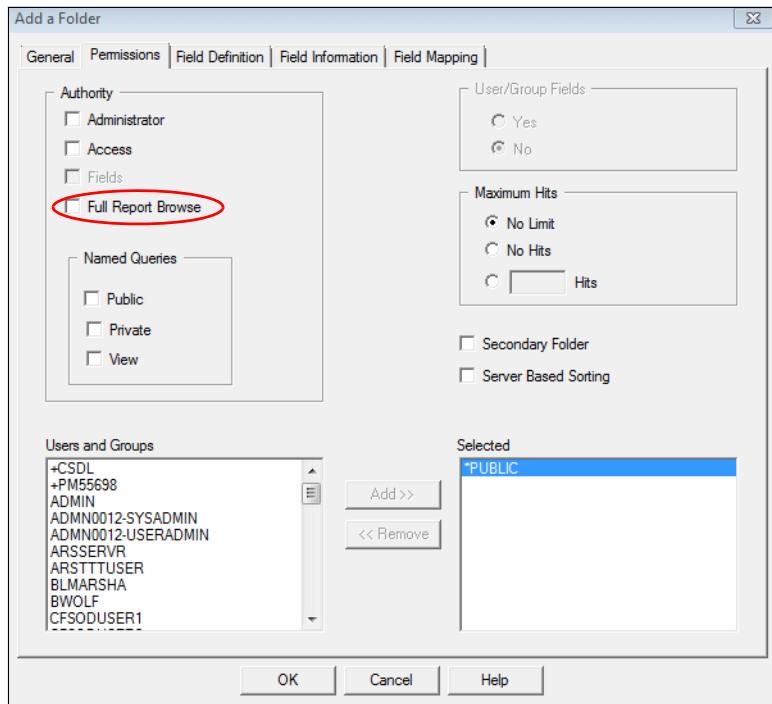


Figure 3-8 Folder Permissions tab

If the user has Full Report Browse authority for a specific folder (which can be configured in the Administrator Client), the Windows client has a new View Full Report button. When the user clicks it, Content Manager OnDemand retrieves the entire report so that the user can view it. If the user does not have Full Report Browse authority, the button is not visible in the Windows client.

If you click **View Full Report**, the entire report (with the same load ID) that is associated with the selected document is viewed, rather than the individual document. If a Full Report document is displayed and the entire document is printed to a server printer, the entire report is printed as a single job.

Maximum Hits

The Maximum Hits section (Figure 3-8) sets the maximum number of document hit list entries that are returned by a folder query. Limiting the number of hits that can be returned from a query prevents performance degradation that might occur if a large result is returned from a query. If a query results in a large hit list that takes a long time to create, the cancel operation function on Content Manager OnDemand Client can be used to stop the creation of the hit list.

Secondary Folder

The Secondary Folder parameter (Figure 3-8) is used to manage the number of folders that a user is presented with when the user logs on to the Content Manager OnDemand system and their list of folders is displayed. By default, Content Manager OnDemand presents a list of the primary folders that a user is authorized to access. Marking a folder as a secondary folder reduces the size of the initial folder list. All folders that the user is authorized to view might be displayed by selecting the show all folders in Content Manager OnDemand Client.

Server Based Sorting option

The Server Based Sorting option (Figure 3-8 on page 56) is used to sort the document hit list on the server before it is returned to the client.

Important: Sorting might still occur on the client if any of the following items are true:

- ▶ Multiple application groups are searched. (The folder contains multiple application groups.)
- ▶ The search query is too long or too complex for a single SQL statement.
- ▶ The user specifies the Append option.

Text Search

Text Search (Figure 3-9) is used to search documents that contain a specific word or phrase before the document hit list is built. Only documents that contain the specified word or phrase are returned as part of the hit list. The search takes place on the server.

Figure 3-9 shows the Text Search option in the Field Definition tab of the Add a Folder window.

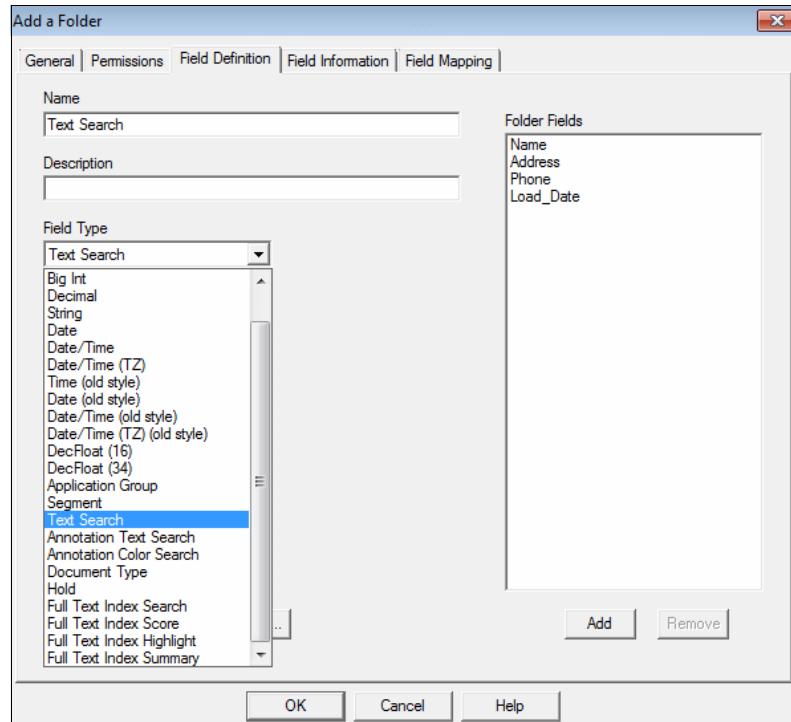


Figure 3-9 Text Search

By using Text Search, a user can further qualify a search without adding the processing that is associated with adding and maintaining additional index fields to the database. Text search is performed on the documents that match the criteria for the other query fields. For example, if the other query fields are date and account number, a text search is performed on the documents that match the specified date and account number. If the document contains the text search string, it is returned as part of the hit list. Text search fields are not mapped to database fields.

A text search string can be a word or a phrase. Only one text search field can be defined per folder. The only valid search operator is EQUAL. Wildcard searches and pattern searches are allowed. Text search is not case-sensitive.

3.1.5 Cabinets

A *cabinet* is a container for folders. You can use cabinets to manage folders and enable users to navigate to folders more easily. A folder can belong to one or more cabinets.

3.1.6 The report wizard

So far, we described how to use Content Manager OnDemand reporting tools to create an application group, an application, and a folder as separate actions. Two methods exist to define a report to Content Manager OnDemand:

- ▶ Add a separate application group, an application, and a folder.
- ▶ Use the report wizard.

This section briefly describes the report wizard's capabilities.

The report wizard defines a report to Content Manager OnDemand by combining the tasks of adding an application group, an application, and a folder into one task. Information for the application, application group, and folder is gathered by answering a series of questions and by using the graphical indexer to define the indexing parameters, the database fields, and the folder fields. Alternatively, database fields and folder fields can be defined without using the graphical indexer.

To start the report wizard, you click the report wizard icon on the main window of the Administrator Client, as shown in Figure 3-10.

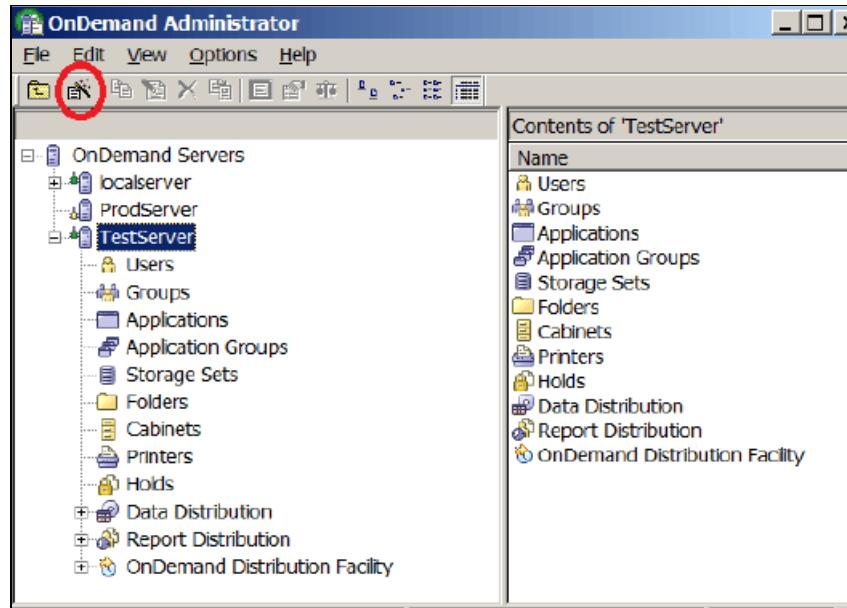


Figure 3-10 Report wizard icon on the OnDemand Administrator Client

Report wizard settings

As you move through the report wizard, standard options are selected for you. Use the defaults unless you have a specific reason not to use them. Depending on how you use the Report Wizard, you might not see all of the windows that we will describe.

Introduction window

The Report Wizard introduction window provides a brief explanation of the report wizard. Your first step is to select the indexer that you want to use to index the data. For all indexers, you specify the type of data that you want to store. For indexers other than Generic and XML, you specify the location of the sample data.

Choose the indexer and type of data and then set up the sample data, as shown in Figure 3-11.

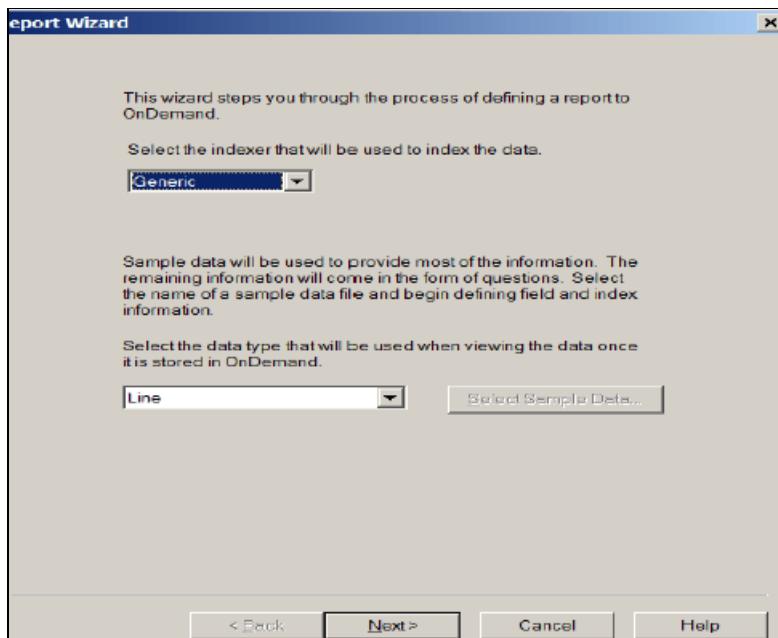


Figure 3-11 Setting up the report wizard

On z/OS or Multiplatform implementations, if AFP is selected as the data type and the report data is line data, it is converted to AFP before it is loaded into Content Manager OnDemand. The report wizard cannot be used to define a report to Content Manager OnDemand if the report is already AFP data.

Report window

The Report window (Figure 3-12 on page 60) displays the sample data and provides easy-to-use tools to help you define indexing information, database fields, and folder fields. Press F1 to display the online help for options and commands that are available on the Report window. Use the online help to learn how to define triggers, fields, indexes, database fields, and folder fields.

Important: When you finish defining the indexing, database, and folder information, save your changes.

The screenshot shows a Windows application window titled "C:\wagner\Samples\Line Data\bike.credit.9801 - Warning! This data should...". The main area contains a grid of data with the following approximate structure:

000-000-00001/01/9802/01/9812000.0011064	0	22	93.57	93.57	93.57	1	
PAYMENT - THANK YOU							
12/01UNITED AIRLINES	DENVER	CO1415236542	18410712/02	82.85			
12/01GART BROTHERS	WESTMINSTER	CO297133158	18190812/02	0.54			
12/01K-MART	ARVADA	CO1186886479	19810102/02	65.14			
12/01WORLD TRAVEL	LOUISVILLE	RY826696551	70934612/02	133.24			
12/01TARGET	PETOSKY	MI124751475	16433512/02	43.17			
12/01SOUND WAREHOUSE	DENVER	CO1765216695	7800112/02	46.42			
12/02CONOCO OIL CO.	NASHVILLE	TN844694334	41664712/03	123.12			
12/02BLACK-EYED PEA	OAK RIDGE	TN2037248455	4301412/03	188.01			
12/02AMOCO	DENVER	CO898644300	11721512/03	197.82			
12/02ROCKIES BASEBALL	LEXINGTON	RY1181603376	1489812/03	5.09			
12/02TARGET	LAS VEGAS	NV1812726020	9819112/03	50.31			

Figure 3-12 Report window

View information window

If you specify User Defined as the data type on Figure 3-11 on page 59, you must specify a file extension. If you specify Line as the data type, you must specify the code page, carriage control (CC), and record format (RECFM). See Figure 3-13.

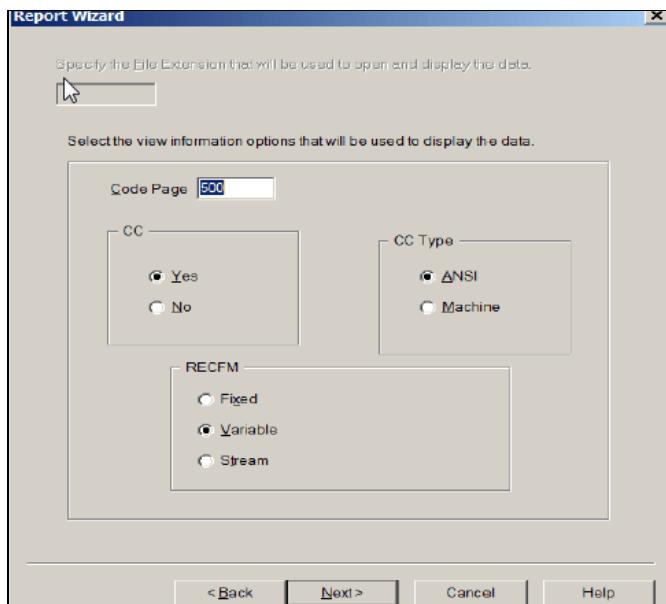


Figure 3-13 View information options

Managing fields window

When you select the Generic indexer or the XML indexer, you add and remove database and folder fields.

When you click Add or Properties (Figure 3-14 on page 61), the report wizard displays a window where you specify the properties of a field.

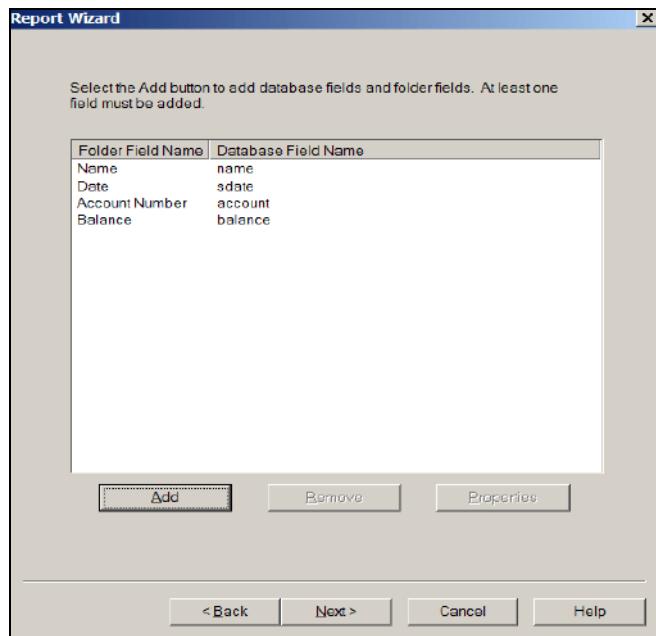


Figure 3-14 Managing fields

Managing data window

When you load a report into the system, you can specify that you want report data to be stored by using large object support. You also need to specify how you want Content Manager OnDemand to manage annotations that users attach to pages of the report. See Figure 3-15.

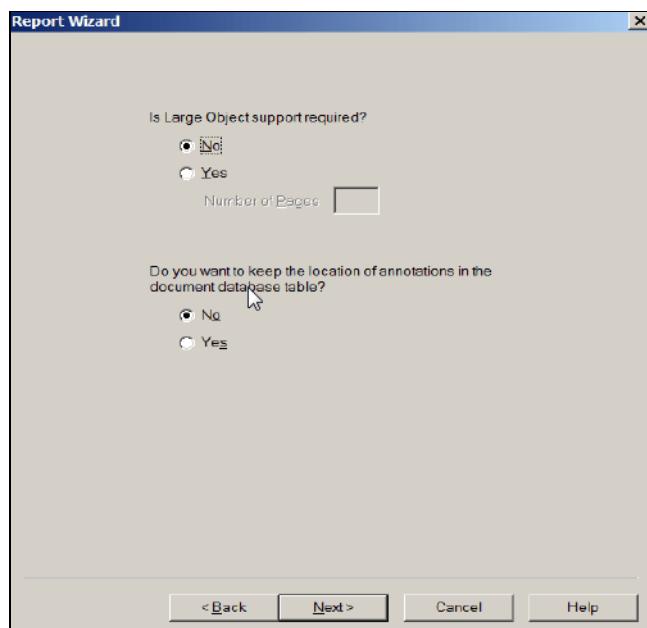


Figure 3-15 Managing data

Application identifier window

When you use the report wizard to add an application to an existing application group, you must specify the name of the application and select a value that uniquely identifies the application within the application group. See Figure 3-16 on page 62.

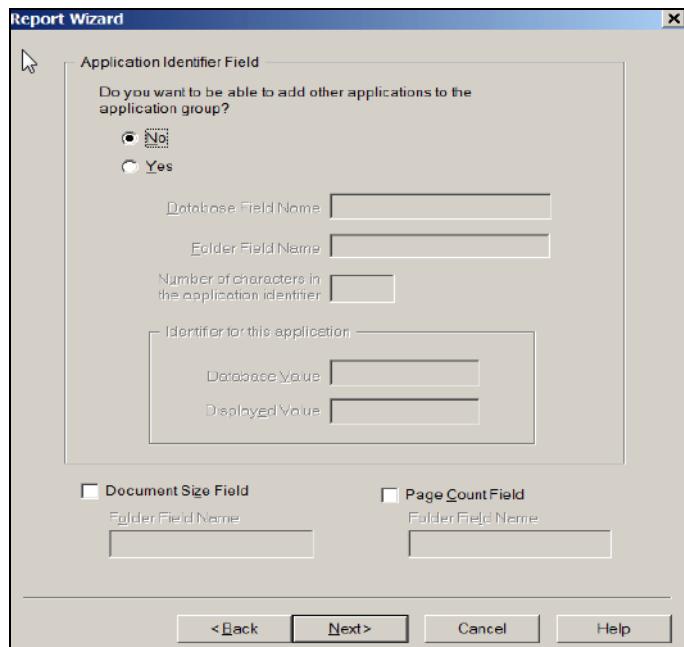


Figure 3-16 Application identifier

Storage management window

The storage management window (Figure 3-17) determines where the storage manager maintains copies of reports and also determines how and when Content Manager OnDemand deletes report data from the system.

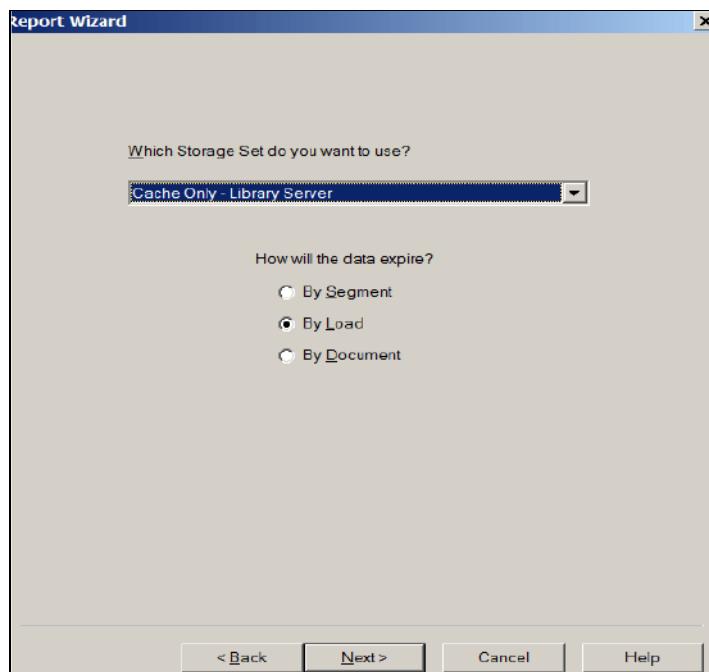


Figure 3-17 Storage management

Applications in the application group window

If the report that you define is one of several reports that will be stored in the same application group, you can use the report wizard (Figure 3-18) to define the following information for the report:

- ▶ The database field that contains the values that identify an application within the application group
- ▶ The folder field that users use to search a specific application
- ▶ The length of the application ID field

If you select Document Size, Content Manager OnDemand adds a field to the application group and folder. Content Manager OnDemand stores the size of the document in the application group field when data is loaded.

If you select Page Count, Content Manager OnDemand adds a field to the application group and folder. Content Manager OnDemand stores the number of pages in the document in the application group field when data is loaded.

You must provide the folder names for both fields (Document Size and Page Count). You do not need to specify the names for the application group fields because they are predefined.

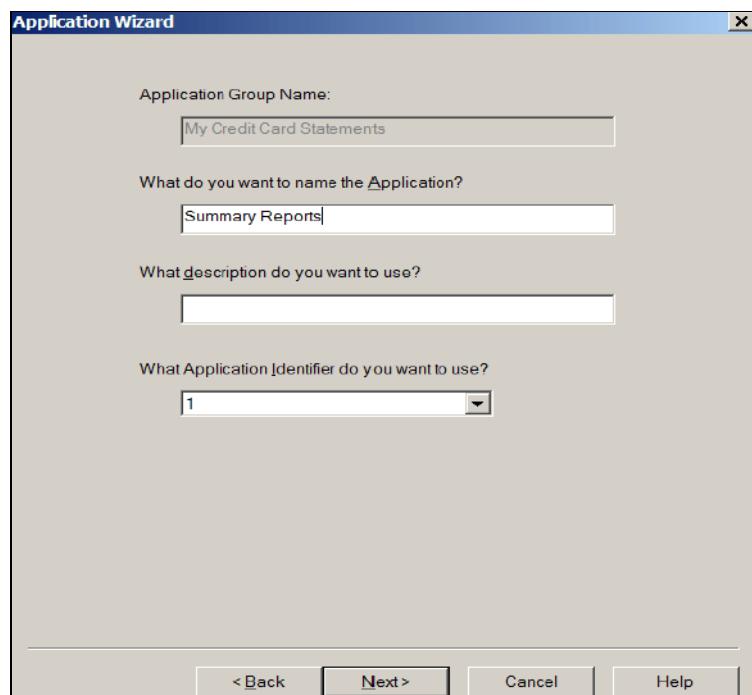


Figure 3-18 Application group

Enhanced Retention Management and Interoperate with IBM FileNet P8 Platform window

In this window (Figure 3-19 on page 64), you can configure the application group to work with the following features:

- ▶ Enhanced Retention Management feature of Content Manager OnDemand
- ▶ Interoperability between Content Manager OnDemand and IBM FileNet® P8 Platform

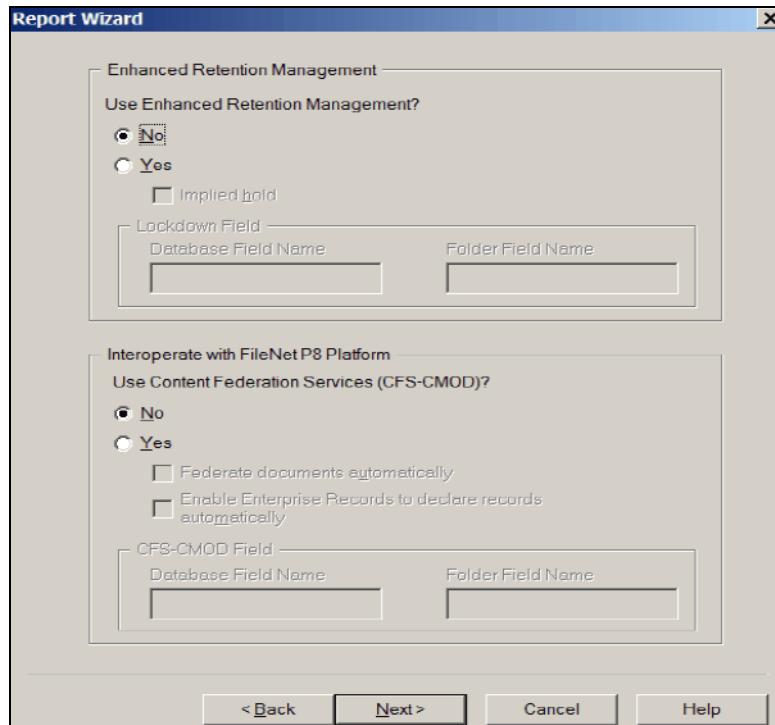


Figure 3-19 Enhanced Retention Management and Interoperate with FileNet P8 Platform window

Full Text Index window

If Full Text Indexer is installed, specify the name of the Content Manager OnDemand Full Text indexing server and optionally, add Full Text Index folder fields (Figure 3-20).

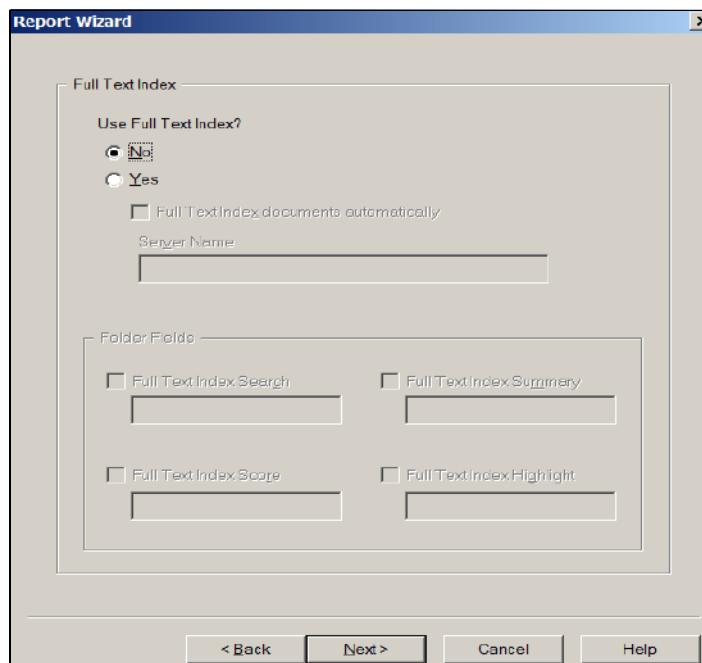


Figure 3-20 Full Text Index window

Name window

In this window (Figure 3-21), specify the names of the application group, application, and folder. After you enter the names, Content Manager OnDemand queries the library server to ensure that the names are valid and unique.

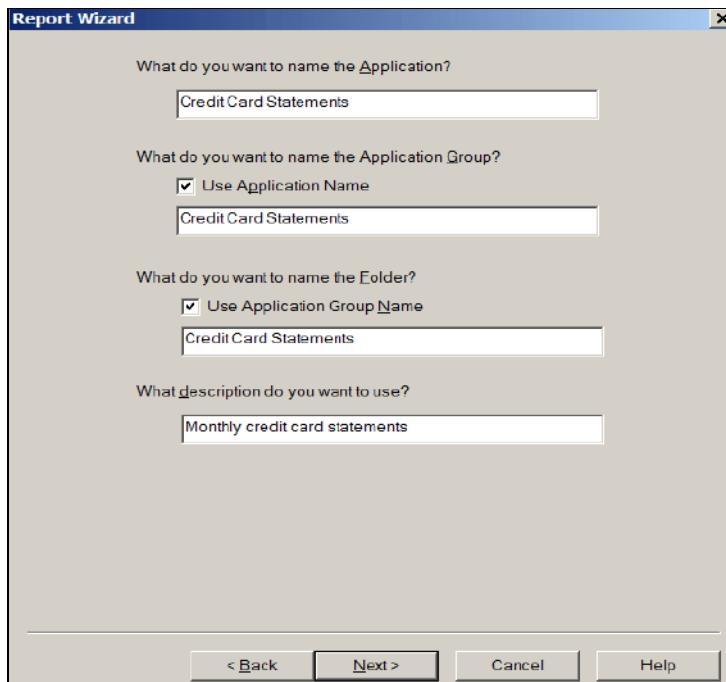


Figure 3-21 Specifying names

Wizard complete window

In this window (Figure 3-22 on page 66), you confirm the selections that you made for the report. Click **Display** to view a summary of the application group, application, and folder definitions. From the summary window, choose the **Print** icon from the toolbar to print a copy of the definitions.

When you are satisfied with the selections that you made for the report, click **Finish** to complete the report definition. Content Manager OnDemand adds the application group, application, and folder to the library server, closes the report wizard, and returns to the administrator window.

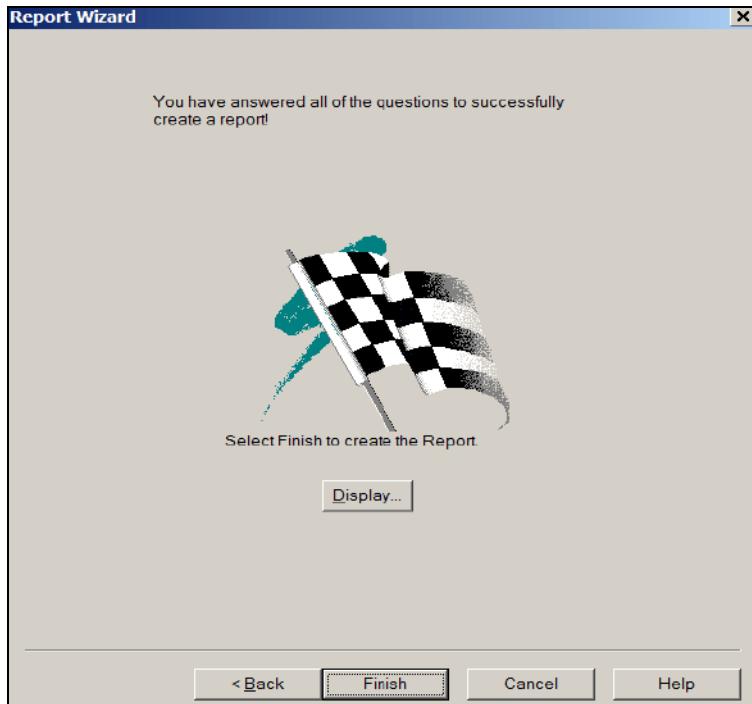


Figure 3-22 Completion window

3.2 User and group administration

When you design a Content Manager OnDemand system, you must determine the best way to implement the many authority structures that are available for users and administrators of your system. The span of control for the administration of the system must be considered with the level of user access to the data that is stored in the system. How many different administrators are required? Will all administrators have system administrator authority or will different administrators have different levels of authority? What is the most effective way to restrict a user's access to only the data that is necessary to do that user's job?

The answers to these questions depend on the size of the system, the degree of centralization to be exercised over system administration, and the nature of the data and the business needs of the users.

Centralized or decentralized

In a system design that exercises centralized control, one or a few administrators are granted system administrator authority. A centralized system typically is used when the number of reports and users to be added to the system is small. Centralized administration is also appropriate where resources are limited and only one person might have the skills and knowledge to perform the system administration tasks, or where one user group performs all of the administration tasks.

In a system design with decentralized control, different users are granted different levels of administrative authority. For example, you might have users that have the authority to create users and groups. Other users might have the authority to create application groups and folders, and others might be given full system administration authority.

The skill level of the users might be a determining factor in the degree of authority that is granted. It takes a more skilled user to define indexes and report parameters than to set up users and groups. A decentralized system is typically used when data from different sources is stored on the same Content Manager OnDemand system but must be maintained independently of other data. Decentralization also makes sense when report loading and processing needs are limited to a specific group of users for security purposes or when administrators that add users and groups must be prevented from accessing report data.

The decision about whether to use a centralized or a decentralized administration model is best made *before* any data is set up in the system. Even though the type of administration that is chosen can be changed later, the amount of work that is involved in that change is greater than the amount of work that is necessary to study the requirements of the system and implement the appropriate administration policies from the beginning.

In this section, we describe different types of users, followed by a description of a decentralized administrative plan. We also introduce a new administrative tool, Content Manager OnDemand XML Batch Administration, which is a command-line program that is run on the Content Manager OnDemand server.

3.2.1 User types, authorities, and functions

Four types of users are available in a Content Manager OnDemand system. Each type has a different level of access, authority, and responsibility in the system:

- ▶ User: Logs in and queries the system to retrieve documents and reports for viewing.
- ▶ User administrator: Adds users or other user administrators to the system.
- ▶ Report administrator: Defines the application groups, applications, folders, and cabinets to be part of the system. The report administrator is responsible for understanding the report and document data and for defining the indexes to be extracted from the data and stored. A report administrator is also responsible for designing the user interface to the reports through the folder definition process and for controlling access authority to the reports that the report administrator designs, indexes, and loads.
- ▶ System administrator: Has the highest level of authority in a Content Manager OnDemand system. The system administrator has authority for all system functions and can grant other users the authority to perform various tasks. The system administrator is the only level of authority that can create storage sets and define system printers.

When the administrative tasks and levels of authorities are understood, you must decide the span of control in the system. Is it better to have one user control all access and functions in the Content Manager OnDemand system, or is it better to spread the administrative tasks among several users to smooth the workload based on system requirements? The answer to this question depends on whether your environment uses centralized or decentralized administrative control.

A centralized administrative plan is best suited for a Content Manager OnDemand system with a few users and relatively few reports to define. In the next section, we focus on the decentralized system and describe the different aspects of a decentralized administrative plan.

3.2.2 System administration

Content Manager OnDemand can centralize or decentralize the administration of the system. A centralized environment means that one type of user, a system administrator, controls the creation and access to all of the objects that are defined on the system. A decentralized environment means that the tasks of the system administrator are divided and assigned to other users. The responsibilities of the other users might vary from user administration, group administration, application group administration, folder administration, or any combination of the administrative tasks.

You need to decide whether to centralize or decentralize the administration of the system before objects are added to the system. Although the decision is reversible, the amount of work that is required to change from one type of administration to the other can be significant if many users, groups, folders, and application groups are already defined to the Content Manager OnDemand system.

Many ways exist to decentralize the administration of the system because of the various user types and the additional authority levels that can be specified for users. Two specific models are described in this section: the object type model and the object owner model.

Object type model

In the object type model, which is shown in Figure 3-23, all of the objects on the system are logically grouped into administrative domains according to the type of the object. The administrator of a domain maintains all of the objects within the domain. For example, an application group, folder, or cabinet administrator maintains all of the application, application group, folder, and cabinet objects on the system.

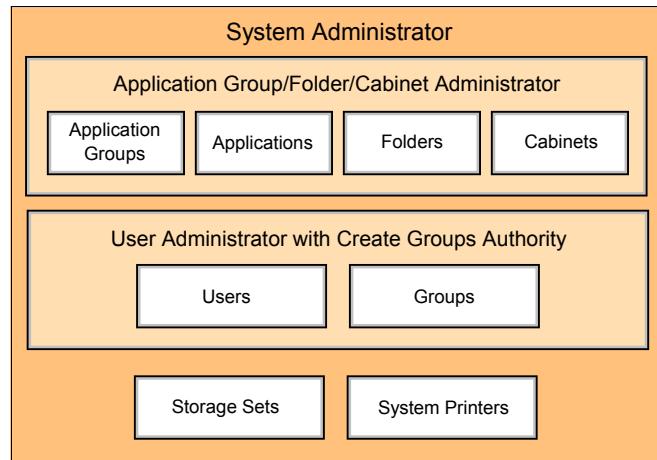


Figure 3-23 Decentralized system administration - object type model

In this model, the system administrator defines two new users. One user is responsible for administering applications, application groups, and folders, and this user is defined as an application group, folder, and cabinet administrator. The second user is responsible for administering users and groups, and this user is defined as a user administrator with the Create Groups authority.

Table 3-1 on page 69 shows the administrative users and the tasks that are assigned to the users.

Table 3-1 Administrator roles in object type model

Administrator type	Administrative tasks
System administrator	Creates report administrators. Creates user administrators with Create Groups authority. Creates and maintains storage sets. Creates and maintains system printers.
Application group, folder, and cabinet administrator	Creates and maintains application groups. Creates and maintains applications. Creates and maintains folders. Creates and maintains cabinets.
User administrator with Create Groups authority	Creates and maintains users. Creates and maintains groups.

When they are maintaining application groups and folders, the application group, folder, and cabinet administrator must give other users access to the application groups, folders, and cabinets. The recommended and simplest way to perform this task is to give access to a group, rather than to individual users. No additional work is required by the application group, folder, and cabinet administrator when another user needs access to the application group, folder, or cabinet. When a new user is added to the group, the user automatically gets access to the application group, folder, and cabinet. Adding the user to the group is the responsibility of the user administrator because the user administrator owns all of the groups in this model.

Another reason for giving groups rather than individual users access to application groups and folders is that the application group, folder, and cabinet administrator does not have access to the users and groups in this model. Because the application group, folder, and cabinet administrator must first be granted access to any users or groups that require access to application groups, folders, or cabinets, it is simpler and less time-consuming to give access to a few groups rather than hundreds or even thousands of users. The application group, folder, and cabinet administrator is given access to a group by adding the application group, folder, and cabinet administrator to the group. This task is performed by the user administrator with the Create Groups authority. As a group member, the application group, folder, and cabinet administrator can see the group in the list and can grant group access to any application groups and folders on the system.

To give an application group, folder, and cabinet administrator access to a user, the user administrator with the Create Groups authority must update each user and give the application group, folder, and cabinet administrator access to the user. After access is granted, the application group, folder, and cabinet administrator can see the user in the list and can grant the user access to any application groups, folders, and cabinets on the system. Again, this approach is not recommended because this task must be repeated each time that a user is added to the system.

Object owner model

In the object owner model, which is shown in Figure 3-24 on page 70, the objects on the system are logically grouped into administrative domains according to the creator or owner of the object. An administrator maintains only the objects that they create. For example, a user with Create Application Groups and Create Folders authority can maintain only the applications, application groups, and folders that they created.

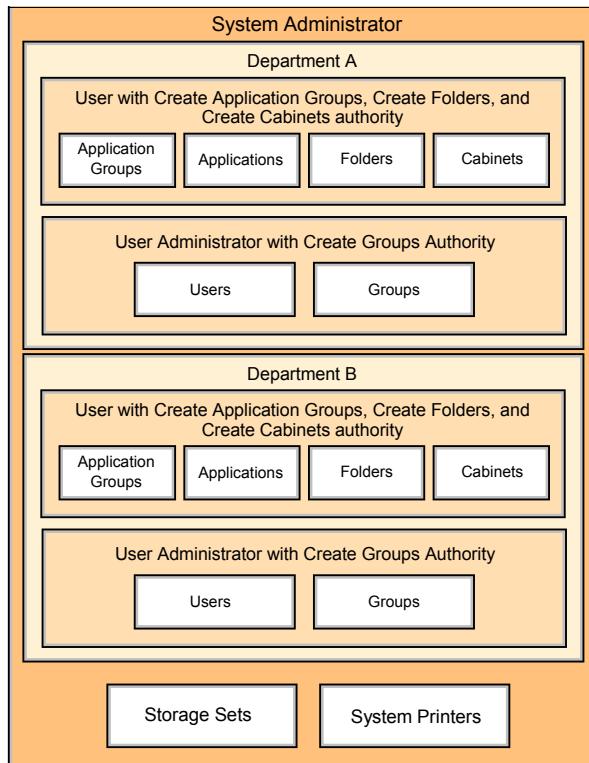


Figure 3-24 Decentralized system administration - object owner model

The object owner model can be used to separate the objects on the system into logical parts, such as a department, company, or another entity. Each part is independent of the other part and each part must be maintained separately. Each part typically requires two administrative users. One user is responsible for creating and maintaining users and groups. The other user is responsible for creating and maintaining applications, application groups, and folders. However, you can also define one user with the authority to create and maintain users, groups, applications, application groups, and folders. In effect, the one user is the system administrator for a logical part of the system.

In the object owner model, the system administrator defines two users for each logical part of the system. One user is responsible for maintaining the users and groups for a logical part of the system. The other user is responsible for maintaining the applications, application groups, folders, and cabinets for a logical part of the system. With the object owner model, you store data from several sources on one Content Manager OnDemand system and let only one set of users access each set of data. Table 3-2 on page 71 shows the administrative users and the tasks that are assigned to the users.

Table 3-2 Administrator roles in the object owner model

Administrator type	Administrative tasks
System administrator	Creates a report administrator with Create Application Groups and Create Folders authority. Creates a user administrator with Create Groups authority. Creates and maintains storage sets. Creates and maintains system printers.
Report administrator	Creates and maintains application groups. Creates and maintains applications. Creates and maintains folders. Creates and maintains cabinets.
User administrator	Creates and maintains users. Creates and maintains groups.

To illustrate how the object owner model can be used, assume that a company installs a Content Manager OnDemand system to provide data archival and retrieval services for other organizations. The company provides the hardware and software that are required to administer the system and archive and retrieve the data. An administrator from each organization defines application groups and folders for their data. Another administrator defines the users that can access the data. The system must be able to limit access to an organization's application groups and folders. Only users that are defined by an organization must have access to the application groups and folders that are owned by the organization. The system must also be able to limit access to the data. Only users that are defined by an organization must have access to the data that is owned by the organization.

Summary

Choosing the correct administration model is an important decision in the design of a Content Manager OnDemand system. Table 3-3 contains general guidelines to consider when you decide on an administration model.

Table 3-3 Administration guidelines

Environment	Recommendation
The number of reports and users to add to the Content Manager OnDemand system is small (fewer than 100).	Centralized system administration
Resources are limited and only one person performs system administrative tasks.	Centralized system administration
All of the system administration tasks are performed by one group.	Centralized system administration
Data from several independent sources is maintained on the same Content Manager OnDemand system. The data must be kept independent of other data in the system. Data must be isolated and access is allowed only for users who must view the data.	Decentralized system administration that uses the object owner model
Report processing and loading must be limited to a group of users for security reasons.	Decentralized system administration that uses the object type model
The administrator that adds and maintains users must not have access to the report data. A separate administrator performs report administration and loading.	Decentralized system administration that uses the object type model

3.3 Content Manager OnDemand XML Batch Administration

In addition to the Administrator Client that runs under Windows, Content Manager OnDemand provides an administrative program that uses Extensible Markup Language (XML). The XML Batch Administration program (XML batch program) is run on the Content Manager OnDemand server and provides the same functionality as the Administrator Client.

The difference between the two programs is that for the Administrator Client, the user must provide input through the graphical user interface (GUI) as opposed to the XML batch program, which receives input through the XML interface.

In this section, we describe the following items:

- ▶ Benefits of using the XML batch program
- ▶ Using the XML Batch Administration program
- ▶ Special features of the XML batch program
- ▶ Tips on using the ARSXML command

Benefits of using the XML batch program

Many benefits are possible when you use the XML batch program:

- ▶ It provides another way to perform the Content Manager OnDemand system administrative tasks.
- ▶ It can process different types of objects, such as updating users in a group and application group permission at the same time.
- ▶ The Administrator Client is not needed.
- ▶ It is useful for replicating the same objects to multiple Content Manager OnDemand servers, and it can even replicate the object when no network connection exists between the servers.
- ▶ It simplifies the automation of system administrative tasks.
- ▶ For Content Manager OnDemand support purposes, the output XML file can be used to provide information to the support team for problem determination.

3.3.1 Using the XML Batch Administration program

This section provides a brief explanation of how to use the new XML batch program. For more information, see *IBM Content Manager OnDemand for Multiplatforms - Administration Guide*, SC19-3352.

The Batch Administration program is called **arsxml**. With this XML batch program, you can export, add, delete, and update Content Manager OnDemand objects.

To use the program, you must have the following files:

- ▶ The schema file, ondemand.xsd
- ▶ An input XML file (for example, exportusers.xml)
- ▶ A password stash file

In XML, the definition and syntax of the markup language are defined in a *schema file*. For the Content Manager OnDemand XML batch program, the schema file is called ondemand.xsd. It contains the definitions for the Content Manager OnDemand objects: users, groups, applications, application groups, storage sets, folders, printers, and others. Each Content Manager OnDemand object definition contains one or more child objects. For example, a user object has a child object for permissions, and a group object has a child object for users in the group. The schema file (ondemand.xsd) must not be changed in any way by the user.

The *input XML file* for the XML batch program is parsed to ensure that it is valid according to the schema file. Each object within the file is examined to ensure that the attributes are valid according to the object type. The XML batch program generates XML when Content Manager OnDemand objects are exported. The XML that is generated can be used as an input for the subsequent **arsxml** command.

Example 3-1 shows a sample of the file exportusers.xml from the XML samples directory. You can change the names of the users to the users that you want to export.

Example 3-1 Sample XML input file for exporting users

```
<?xml version="1.0" encoding="UTF-8"?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="../ondemand.xsd">

    <user name="SAMPLEUSER0" />
    <user name="SAMPLEUSER1" />
    <user name="SAMPLEUSER2" />
    <user name="SAMPLEUSER3" />
    <user name="SAMPLEUSER4" />
</onDemand>
```

You can export objects by running **arsxml export**. The following command exports the users that are listed in the exportuser.xml file, from the server odserver1, to an output file named users.xml:

```
arsxml export -u oduser1 -p /my/stash/pwfile -h odserver1 -i exportusers.xml -o
users.xml -v
```

You can import objects by running **arsxml add**. The following command imports the users from the users.xml file (which is generated from the previous command) to server odserver2:

```
arsxml add -u oduser2 -p /my/stash/pwfile -h odserver2 -i users.xml -v
```

You can delete objects by running **arsxml delete**. The following command deletes the users from odserver2, based on the users that are listed in the users.xml file:

```
arsxml delete -u oduser2 -p /my/stash/pwfile -h odserver2 -i users.xml -v
```

For deletion, you are prompted before each object in the XML is deleted, unless the **-x** parameter is used.

You can update objects by running **arsxml update**. For example, you want to update the description of the user User1 with a new description and add the authority to create users. In this case, you construct the XML input file that is shown in Example 3-2.

Example 3-2 Input file to update user - updateUser.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ondemand.xsd" >

    <user name="User1" description="User1" createUsersAuth="Yes" >
        </user>
    </onDemand>
```

The following command updates user User1:

```
arsxml update -u oduser2 -p /my/stash/pwfile -h odserver2 -i updateUser.xml -v
```

Certain attributes are not allowed to be updated, such as the data type of an application group field or folder field. If you specify to ignore and continue, the XML batch program produces a warning message and the rest of the attributes continue to be updated.

You can validate the input XML file by running **arsxml validate**. When the validate command is used, only the lines in the input XML file are checked. No call to the Content Manager OnDemand server is made. The following command validates the input XML file:

```
arsxml validate -i sample.xml
```

Note: When you create an input XML file, not all attributes must be specified for each object.

3.3.2 Special features of the XML batch program

You can add user or group permissions to an application group or folder by adding a permission child object to the application group or folder group object.

Dependent objects, such as all users that belong to a group, can be exported together when you choose to export the group rather than having to add a user object to the XML file for every user in the group. You export the group by specifying the **arsxml** command option **-r d** on the command line.

In a case when no network connection exists between two servers, the XML batch program can be used to export Content Manager OnDemand objects to an XML file from one server and later import to another server.

If an error occurs during the processing of one of the objects in the input XML file, the remainder of the XML file is not processed unless option **-e c** is used on the **arsxml** command line.

Note: Objects must be specified in the correct order. For example, when you add application groups and applications in the same XML file, you must first specify all of the application groups and then specify the applications.

3.3.3 Tips on using the ARSXML command

If you are not familiar with the syntax of the **ARSXML** command, an easier way to begin is to perform an export of the object. By doing so, you get a working XML input file that you can modify to suit your needs. Ensure that the export is successful without any errors; otherwise, the output XML file might be incomplete.

Adding objects to the Content Manager OnDemand server is straightforward. If you are performing more advanced operations, such as updating the permission of an existing user for an application group or folder, and you are not getting the results that you are expecting, the **task** attribute might be missing. You must include this attribute when you want to update an existing object, such as removing a user's permission from an application group or updating a user's permission to an application group. The values for the task attribute are add, delete, and update.

For example, if you want to remove the permission of the user User1 from an application group, you must use the following line in the input XML file:

```
<permission user="User1" task="delete" />
```

Another example is when you want to update the query restriction of the user User1 for the application group CreditCardAG. In this case, you must use the following line in the input XML file, with the task attribute set to update.

```
<permission user="User1" task="update" queryRes="account='000-000-000'" />
```

The previous line is incorporated in Example 3-3 for the input file updateag.xml.

Example 3-3 Input file updateag.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="../ondemand.xsd">

    <!-- update application group with query restriction-->
    <applicationGroupname="CreditCardAG" >
        <permission user="User1" task="update" queryRes="account='000-000-000'" />
    </applicationGroup>

</onDemand>
```

The following command updates the query restriction for user User1:

```
arsxml update -h odserver -i updateag.xml -v -u User1 -p /my/stash/pwfile
```

Example 3-4 shows the output from the previous command.

Example 3-4 Successful output of updating the query restriction for user User1

```
ARS6822I Attempting login for userid 'User1' on server 'odserver'
Updating applicationGroup, CreditCardAG
Update of applicationGroup, CreditCardAG was successful.
Updating applicationGroup-permission, CreditCardAG-User1
Update of applicationGroup-permission, CreditCardAG-User1 was successful.
Finished processing file updateag.xml.
```

The operation is successful. If you do not specify task="update" in the input XML file, you see a message that indicates that the object exists, as shown in bold in Example 3-5. In this scenario, user User1 is not updated with the new query restriction.

Example 3-5 Output of updating the user without using task="update"

```
ARS6822I Attempting login for userid 'User1' on server 'odserver'Updating  
applicationGroup, CreditCardAG  
Update of applicationGroup, CreditCardAG was successful.  
An applicationGroup-permission object named 'CreditCardAG-User1' already exists.  
Finished processing file updateag.xml.
```



Database structure

In this chapter, we describe the IBM Content Manager OnDemand (Content Manager OnDemand) database structure and relationships between the tables. We list the system control tables and the important data table structures. We explain the relationship between the tables when you load data, show how a search is performed on the database tables, describe the system log, and describe special considerations for DB2 on z/OS.

In this chapter, we cover the following topics:

- ▶ System control tables
- ▶ Main data table structures
- ▶ Relationship between tables when data is loaded
- ▶ Search sequence
- ▶ System log
- ▶ Database creation and relationships on z/OS

4.1 System control tables

Content Manager OnDemand creates and uses two sets of tables: a set of system control tables and a set of application group data tables. All system control tables are created by the **arsdb** command (except for the Archive Storage Manager (ASM) tables that are used by Content Manager OnDemand for IBM i server). The application group data tables are created when you load data (reports and documents) into the Content Manager OnDemand system.

Table 4-1 shows the Content Manager OnDemand system control tables with their descriptions.

Note: For a Multiplatform or z/OS server, the complete table name consists of the owner name, which can be the database name or the instance name, and the table name. For example, the application group table ARSAG that belongs to the ODARCH instance has a complete table name of ODARCH.ARSAG.

For the IBM i server, the complete table name is in the format of library/table, where the library name is always the same as the instance name. For example, the application group table ARSAG that belongs to the default QUSROND instance has a complete table name of QUSROND/ARSAG.

Table 4-1 Content Manager OnDemand system control tables

Table name	Purpose	Description
ARSAG	Application group table	One row for each application group
ARSAG2FOL	Field mapping table	One row for each application group field that is mapped to a folder field
ARSAGFLD	Application group field table	One row for each field that is defined in an application group
ARSAGFLDALIAS	Application group field alias table	One row for each database (internal) and displayed (external) value in an application group
ARSAGINDEX	Application group composite index table	One row for each composite index on application group fields
ARSAGPERMS	Application group permissions table	One row for every user that is granted specific permission to an application group
ARSANN	Annotation table	One row for each annotation that is added to a database
ARSAPP	Application table	One row for each application that is defined to Content Manager OnDemand
ARSAPPUSR	User logical views table	One row for each logical view that is defined for a specific user
ARSCAB	Cabinet table	One row for each cabinet that is defined to Content Manager OnDemand
ARSCAB2FOL	Cabinet to Folder table	One row for every cabinet that is defined for a folder
ARSCABPERMS	Cabinet permissions table	One row for every user that is granted specific permissions to a cabinet

Table name	Purpose	Description
ARSCFSODWORK	Catalog of work between Content Manager OnDemand and Content Federation Services for Content Manager OnDemand table	One row for the catalog of work between Content Manager OnDemand and Content Federation Services for Content Manager OnDemand
ARSFOL	Folder table	One row for every folder that is defined in Content Manager OnDemand
ARSFOLFLD	Folder field table	One row for every folder field that is defined for a folder
ARSFOLFLDUSR	Folder user field table	One row for every field that is provided for a user that is granted specific field information for a folder
ARSFOLDPERMS	Folder permission table	One row for every user that is granted specific permissions to a folder
ARSFTIWORK	Full text search work table	One row for every application group for full text search
ARSGROUP	Group table	One row for each group that is defined to Content Manager OnDemand
ARSHOLD	Hold table	One row for every hold that is defined in Content Manager OnDemand
ARSHOLDMAP	Catalog of documents to hold table	One row for every catalog of documents to hold
ARSHOLDPERMS	Hold permissions table	One row for every catalog of hold permissions
ARSHOLDWORK	Hold work table	One row for every catalog of hold work
ARSLOAD	Load table	The load_ID table
ARSNAMEQ	Named query table	One row for each private and public named query that is defined to Content Manager OnDemand
ARSNODE	Node table	One row for each storage node that is defined
ARSPRT	Printer table	One row for each printer that is defined in Content Manager OnDemand
ARSPROPTS	Printer options table	One row for each printer option
ARSPRTUSR	Printer user table	One row for each user that has access to a specific printer
ARSRES	Resources table	One row for each resource ID
ARSSEG	Segment table	One row for each segment of application group data
ARSSET	Storage set table	One row for each storage set
ARSSYS	System parameters table	One row for the entire system
ARSUSER	User table	One row for each user that is defined to Content Manager OnDemand

Table name	Purpose	Description
ARSUSRGRP	Users in group table	One row for each user that is assigned to a Content Manager OnDemand group
ARSUSRGRPID	User group ID table	Maintains the association of users with user owners and their authority for groups
Dynamic name	Application group data table	One row for each document that is stored in the application group

Important: *Do not update* the tables by using SQL commands or DB2 system tools, such as SQL Processor Using File Input (SPUFI) or any other tools. The tables must be updated only by the Content Manager OnDemand Administrator Client or Content Manager OnDemand commands.

4.2 Main data table structures

The Content Manager OnDemand data tables can grow rapidly. You must understand the structure of the data tables and the relationships between them.

Two important tables exist that you must examine here: the *segment table* (ARSSEG) and the *application group data table* (ag_internal_id). The segment table contains one row for each segment of each application group data table. Table 4-2 shows the first four columns of the ARSSEG table structure.

Table 4-2 ARSSEG table structure

Column name	Description
agid	Application group ID
table_name	Application group segment table name
start_date	Segment start date
stop_date	Segment stop date

The ARSSEG table points to the application group data table name (second column of the table, table_name). The application group data table is created or updated during the **arsload** process. The application group data table contains a row for each item that is stored in the application group.

The name of the application group data table is ag_internal_id, which is the identifier that Content Manager OnDemand assigns to the application group when the application group is created with the Administrator Client. The three-digit application group identifier is listed in the Storage Management window of the Administrator Client, as shown in Figure 4-1 on page 81. In this case, the application group identifier is WBA, AGID 5185.

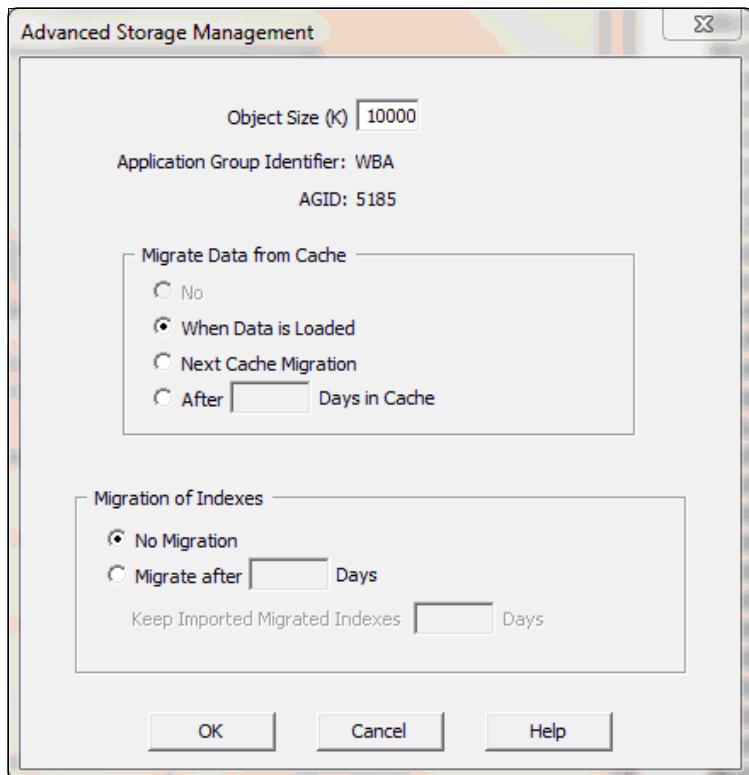


Figure 4-1 Application group identifier example

Table 4-3 shows the first five columns of the application group data table structure.

Table 4-3 AG_internal_id table structure

Column name	Data type	Size	Index	Description
field_1	Varies	Varies	N	First user-defined field in the application group.
field_n	Varies	Varies	N	Last user-defined field in the application group. You can have up to 128 index fields that are defined in Content Manager OnDemand.
doc-name	varchar	11	Y	Document name (object name).
doc_off	integer	4	N	Document offset in the object.
doc_len	integer	4	N	Document length.

The application group data table is indexed on one or more of the user-defined fields, from field_1 to field_n.

Four major factors influence the amount of storage that is needed for the Content Manager OnDemand database:

- ▶ The number of index and filter fields
- ▶ The size of the index and filter fields
- ▶ The number of indexed items per month
- ▶ The number of months (years) Content Manager OnDemand keeps the indexes in the database

Three more tables might grow rapidly during the lifetime of a Content Manager OnDemand system:

- ▶ The annotation table (ARSANN) grows in proportion to the volume of the annotations that are added to the documents. The system creates one row for every annotation. Therefore, every yellow sticker and every graphical annotation add one row to this table.
- ▶ The resource table (ARSRES) grows in proportion to the volume of AFP data that is archived and the resources' (such as formdef, page segments, and overlays) frequency of change.
- ▶ The load table (ARSLOAD) grows in proportion to the number of **arsload** jobs/tasks that are run. The Content Manager OnDemand system creates one row for each load job/task that is run. The load table (see Table 4-4) can grow to a multimillion row table during the lifetime of a Content Manager OnDemand system.

Table 4-4 shows the first four columns for the ARSLOAD table structure.

Table 4-4 ARSLOAD table structure

Column name	Data type	Size	Index	Description
agid	integer	4	Y	Application group identifier
pri_nid	smallint	2	N	Primary storage node identifier
sec_nid	smallint	2	N	Secondary storage node identifier
name	varchar	11	Y	Name of the load

4.3 Relationship between tables when data is loaded

In this section, we present an example that shows the relationships between the Content Manager OnDemand tables when you are loading data to a Content Manager OnDemand system. This example is based on a check application that has four index fields that are defined as *customer_name*, *account_nbr*, *check_nbr*, and *balance*. A one-to-one relationship exists between the application group and the application.

After the application group and the application are defined, the application group gets an application group identifier, *agid*, in the ARSAG table and an internal application group identifier, *agid_name*. Figure 4-2 on page 83 shows the data that is created in the ARSAG table; the *agid* is 5018, and the *agid_name* is HAA.

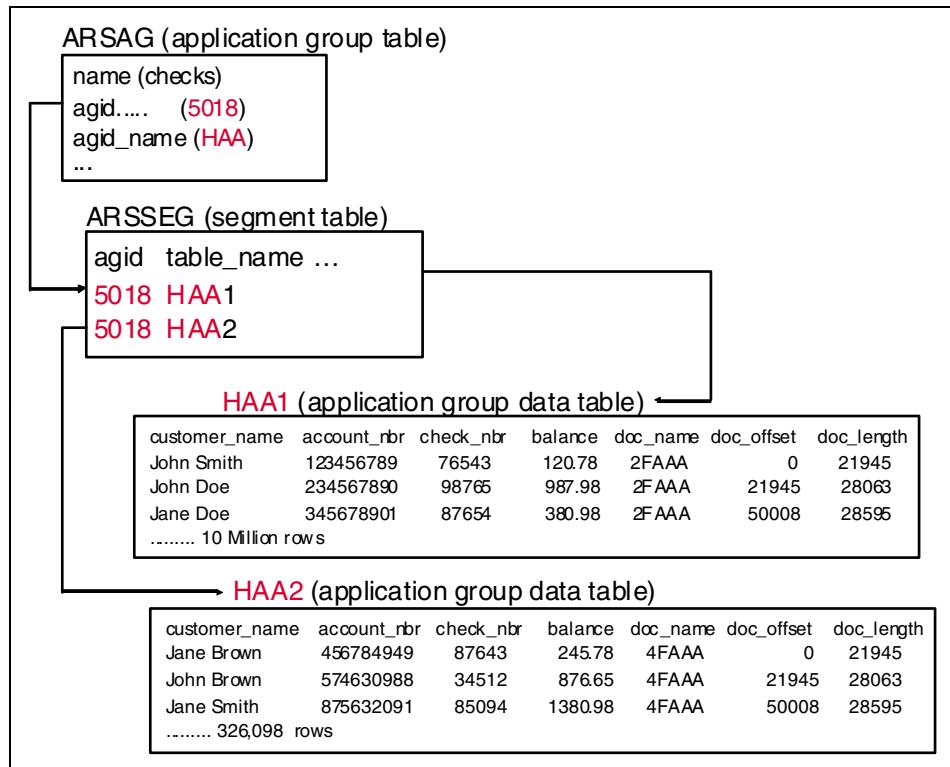


Figure 4-2 Relationship between system tables and data tables

This application group creates an application group data table every 10 million rows (based on the seg_rows value in the ARSAG table, which is not shown in Figure 4-2). During the data loading process, Content Manager OnDemand uses the agid and the agid_name to add a row into the segment table (ARSSEG) for every 10 million rows that are created in the application group data table. When the first data load occurs into the HAA application group, the index values for the stored documents are inserted into table HAA1. A row exists for each document that is loaded. When table HAA1 reaches its max_rows value (in this case 10 million rows), table HAA1 is closed and table HAA2 is opened.

The important pointer in the ARSSEG table is the name of the application group data table, *table_name*, where the index values (in this case, the four defined index values) are stored. The *table_name* consists of the agid_name from the ARSAG table, plus a counter.

Figure 4-2 shows the two rows that are created in the ARSSEG table: one row with the *table_name* HAA1 and another row with the *table_name* HAA2. Both HAA1 and HAA2 are the actual names of the application group data tables that are created.

The application group data table contains the *doc_name*, which is the actual container (storage object) for the individual document. The offset and the document length are also kept in this table. Figure 4-2 shows that the first row has an offset of 0, and the second row has an offset of the document length of the first row.

Figure 4-2 shows the relationship between the tables.

The architecture of relating one application group to one or more application group data tables allows Content Manager OnDemand an unlimited growth of index space. The maximum table size is a limitation of the database subsystem and must be configured for optimal performance.

Because this architecture enables a system to create tables when the maximum table size is reached, no logical limitation exists to the system; rather, the limitation is on the physical resources, such as processing power, disk space, object servers, and storage hardware.

4.4 Search sequence

To better understand the relationship between the various Content Manager OnDemand tables, we describe a search sequence within a Content Manager OnDemand system in this section. A search sequence scans through multiple Content Manager OnDemand tables. We describe the logical flow that the system goes through during a Content Manager OnDemand search.

By using the Content Manager OnDemand standard Windows client, you can open a search criteria window (see Figure 4-3). In our example, four index fields exist: Name, Account, Statement Date, and Balance. The example shows a search for a specific date and balance amount.

The figure shows a screenshot of the Content Manager OnDemand Client search criteria window. The window has a title bar 'Search Criteria'. Inside, there are four search fields:

Field	Operator	Value
Name	Like	<input type="text"/>
Account	Like	<input type="text"/>
Statement Date	Equal To	<input type="text"/> 1994-03-07 <input type="button"/>
Balance	Equal To	<input type="text"/> 104.18

Figure 4-3 Content Manager OnDemand Client search criteria window

After you enter these values, Content Manager OnDemand uses the date information and searches the segment table ARSEG to find the application group data table that contains that date. Content Manager OnDemand then searches the identified table_name (in our example HAA1) for the index values (1994-03-07 and 104.18) and finds the matching *Statement_date* and the *Balance* and returns these values to the client in a search result list.

Any individual document from within this result list can then be retrieved for display on the client. Content Manager OnDemand locates the document in the archive by using the object name, document offset, and length. In the background, the document data is automatically decompressed before it is displayed.

Figure 4-4 on page 85 shows the details of this search sequence from a folder.

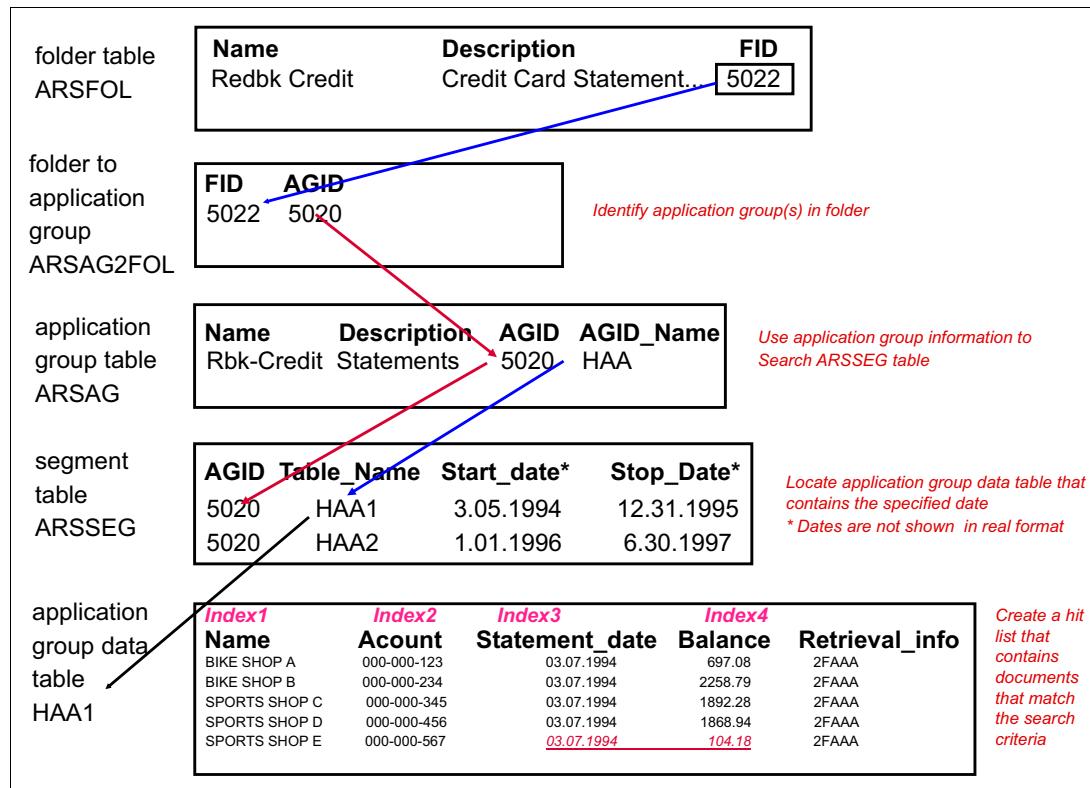


Figure 4-4 Search sequence from a folder

4.5 System log

The system log is used to track all activities in the Content Manager OnDemand Instance. Examples of these activities include logon and document retrieval. The system log is created as an application group. It is created by using the **ARSSYSCR** program. The application group identification name is SL and a 4-byte agid is added. You will see SLXX in the ARSEG table, depending on how large your system log is growing. The creation of a new system log table is based on the number of rows on the Storage Management setting. The default is 10 million rows. This configuration can be modified.

4.6 Database creation and relationships on z/OS

The database creation, allocation of space for tables, and table space of the Content Manager OnDemand product are different in the z/OS environment. The database administrator (DBA) is responsible for the allocation, creation, maintenance, backup, and recovery of the database subsystem.

4.6.1 System tables for Content Manager OnDemand z/OS

For the Content Manager OnDemand z/OS DB2 database environment, standard database backup and recovery procedures can be used for the databases, table spaces, and tables that are created by Content Manager OnDemand. To minimize the effort of creating and monitoring the Content Manager OnDemand data tables, several automated table creation and space allocation procedures are part of the product.

All system tables are created by the **arsdb** program. Each table is created in its own table space. During the installation, the table space is created by member ARSTSPAC in dataset V9R5M0.SARSINST. The size of each table space is specified in dataset V9R5M0.SARSINST. Before you run **ARSTSPAC** to create the Content Manager OnDemand table spaces, you must create the storage group and the database. The **CREATE** for the storage group and database is in member ARSDB2 in dataset V9R5M0.SARSINST. The owner of the database (the submitter of the job or the user ID that is set by the “Set current SQLID = ‘username’”) must match the entry SRVR_INSTANCE_OWNER in the **ARS.INI** file.

The **arsdb** program provides an interface between the database manager and Content Manager OnDemand. Several parameters are used in the creation and dropping of tables. For more information about **arsdb**, see the *IBM Content Manager OnDemand for z/OS - Configuration Guide*, SC19-3363.

The **arsdb** program is in the UNIX System Services file system /usr/lpp/ars/V9R5M0/bin. When you create the Content Manager OnDemand tables or indexes, the **arsdb** command can build the tables and indexes in the default table space or in table spaces that you create by using the ARSTSPAC member.

When you run the **arsdb** command, Content Manager OnDemand validates the existence of the table space. If the table space does not exist, the **arsdb** command creates the Content Manager OnDemand system tables in the default table space. After you create the Content Manager OnDemand table spaces, if changes are required, the best way to change the table space is by running the **ALTER TABLESPACE** command.

4.6.2 Data tables for Content Manager OnDemand z/OS

The data tables in Content Manager OnDemand are created under the control of DB2 on z/OS. Like the system tables, the data tables in Content Manager OnDemand are created dynamically during the **arsload** process. It is important to understand how Content Manager OnDemand on z/OS allocates space for these tables because they can grow large.

During the creation of an application group, a parameter limits the maximum rows for a data table. If this limit is reached, Content Manager OnDemand creates another data table during the **arsload** process. By using the Administrative Client, you can set the maximum row value for an application group data table. This value is on the Advanced section of the General tab in the application group configuration. The field is called *Maximum Rows*.

The space allocation is performed automatically. No further action needs to be performed by the DBA except to set up the backup of the newly created tables and plan for the new resources that are needed for the next couple of months.

For more information about space requirements, see the *IBM Content Manager OnDemand for z/OS - Introduction and Planning Guide*, SC19-3651.

Content Manager OnDemand for z/OS allocates its table space during the creation of a new table based on the following space allocation parameters:

- ▶ DBSIZE/two for primary allocation
- ▶ Primary allocation/four for the secondary allocation

The allocation of the database is in kilobytes. The allocation values depend on the maximum row limit that is set when you create the application group. The DBSIZE depends on the number of index fields that are defined in the application.

DBSIZE is calculated in the following way:

Maximum number of rows * Default Table Factor / records per page

The Default Table Factor is set to “1,2” by the program. The records per page value is a DB2 parameter. For more information about records per page, see Chapter 8, “Estimating Disk Storage”, in the *DB2 Version 10 for z/OS Administration Guide*, SC19-2968.

Note: Based on this calculation, when you define the application group, ensure that you select the appropriate number for Max_Rows:

- ▶ If you expect the number of documents and indexes that are stored to be low, reduce the default value of 10 million rows.
- ▶ If you expect the number of documents and indexes that are stored to be high, increase the default of 10 million rows.
- ▶ If you leave the 10-million-row default unchanged, Content Manager OnDemand allocates 6 million rows as the primary allocation.



Storage management

In this chapter, we explore the storage management options that are available to different IBM Content Manager OnDemand (Content Manager OnDemand) platforms. Content Manager OnDemand can manage the usage of local disk-based storage or cache. Additionally, it supports the usage of various Archive Storage Managers (ASMs) that support external storage devices. These devices are used to manage long-term copies of data but with the development of new disk-based archive devices, different options are now available to users of Content Manager OnDemand.

In this chapter, we cover the following topics:

- ▶ Content Manager OnDemand cache storage
- ▶ IBM Tivoli Storage Manager for Multiplatforms
- ▶ Object access method for z/OS
- ▶ Archive Storage Manager for IBM i

5.1 Content Manager OnDemand cache storage

Content Manager OnDemand has a built-in cache storage management that is used to store documents on locally mounted disk subsystems. These subsystems can be network-attached storage (NAS), storage area networks (SAN), or any type of locally addressable disk that is available to the supported operating system. The cache storage manager uses a list of directories or file systems that are available to determine where space is available for storing and maintaining documents.

Each Content Manager OnDemand object server in the system has a defined set of cache storage devices on which you can maintain the report data for a period to provide the fastest access times for system users.

Certain implementations of Content Manager OnDemand use an all cache system to maintain data for its full retention. Other implementations store to both cache and archive storage. Other implementations store only to the archive.

You can configure Content Manager OnDemand so that at load time one of the following methods of data storage occurs:

- ▶ Data is stored in cache and later is automatically migrated from the cache subsystem to an archive system.
- ▶ Data is stored to both local cache and archive storage.
- ▶ Data is stored directly to archive storage.

These options are described in the following sections.

5.2 IBM Tivoli Storage Manager for Multiplatforms

Content Manager OnDemand for Multiplatforms integrates with Tivoli Storage Manager and a license for this usage is included with Content Manager OnDemand. Within Tivoli Storage Manager, documents can be archived on various media, such as disk, optical, tape, and content-addressable storage (CAS) devices. These archive storage devices must be defined to the Tivoli Storage Manager system. Content Manager OnDemand uses the archive application programming interface (API) that is provided by Tivoli Storage Manager to store and retrieve documents.

To store application group data to the Tivoli Storage Manager ASM, the application group must be configured within Content Manager OnDemand to a defined storage set. This storage set contains a storage node that is defined within Tivoli Storage Manager and points to a specific storage area or media.

With the application group definition, you can specify whether and when the data is migrated to archive storage. For example, you can specify that the data will be migrated to archive storage when the document is originally loaded into the system, or that the data migration occurs the next time that the migration maintenance process is run, or that the data migration occurs after a certain number of days pass from the date that the data was loaded; or never.

In this section, we describe the following two scenarios:

- ▶ The steps that you follow to set up and configure Tivoli Storage Manager as the archive manager for a Content Manager OnDemand for Multiplatforms system.
- ▶ The configuration of IBM System Storage® Archive Manager to store Content Manager OnDemand data. It provides data retention policies that help meet regulatory requirements and uses storage devices, such as EMC Centera or NetApp SnapLock. You must verify that a particular device is supported on the platform of choice.

Starting with Tivoli Storage Manager V6.1, Tivoli Storage Manager uses DB2 for its database instead of the built-in B-tree database. Typically, the Tivoli Storage Manager Server is installed on a separate system. However, for smaller implementations, the Tivoli Storage Manager server can coexist on the same system as your Content Manager OnDemand object server.

5.2.1 Tivoli Storage Manager overview

Before we describe the configuration process, we describe a few of the components that make up a Tivoli Storage Manager system. For a complete description of Tivoli Storage Manager, see the appropriate Tivoli Storage Manager documentation. For example, on Microsoft Windows, see the *Tivoli Storage Manager for Windows Administrator's Guide* V6.3.4, SC23-9773.

Figure 5-1 represents a typical Tivoli Storage Manager system. A short description of each component follows.

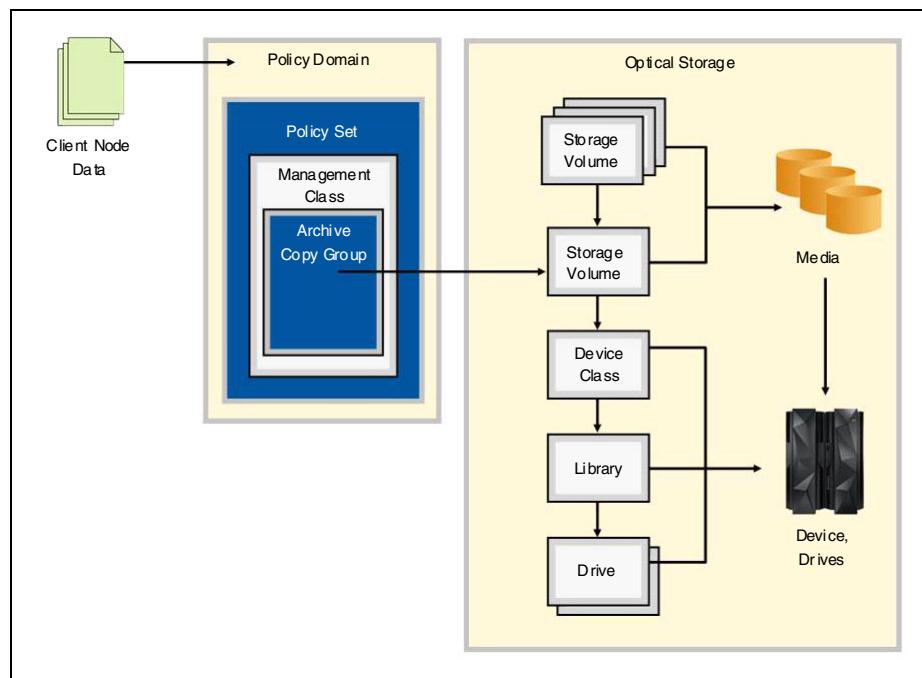


Figure 5-1 Tivoli Storage Manager storage objects

Client node

In Figure 5-1, the client node represents a Content Manager OnDemand object server with an installed Tivoli Storage Manager archive API. The client node is assigned to a policy domain. Each Content Manager OnDemand system that stores reports in Tivoli Storage Manager needs at least one defined client node.

Storage policy

A storage policy consists of the following items:

- ▶ Policy domain: Contains the policy set, management class, and archive copy group that is used by the client node
- ▶ Policy set: Contains management classes, which contain the archive copy groups
- ▶ Management class: Determines where data is stored and how it is managed
- ▶ Archive copy group: Used to copy data to Tivoli Storage Manager for long-term storage

Storage devices and media

Storage devices and media consist of the following items:

- ▶ Library: One or more drives with similar media mounting requirements. Only defined when you have an external library.
- ▶ Drive: A drive mechanism, which is defined by Tivoli Storage Manager, that is in an optical or tape library or stand-alone device.
- ▶ Device class: Specifies the device type and how the device manages media.
- ▶ Storage pools and volumes: A named collection of storage volumes of the same media type that is associated with a device class.

Tivoli Storage Manager installation

For help with installing and configuring IBM Tivoli Storage Manager Version 7.1.1 for Windows, see the installation guide at the following website:

[ftp://public.dhe.ibm.com/software/products/TSM/current/b_srv_install_guide_windows.pdf](http://public.dhe.ibm.com/software/products/TSM/current/b_srv_install_guide_windows.pdf)

By using this guide, complete the steps that are listed to install the Tivoli Storage Manager server, Tivoli Storage Manager licenses, Tivoli Storage Manager backup archive client, and Tivoli Storage Manager Device driver.

When these installations are complete and the Tivoli Storage Manager Server is running, go to 5.2.2, “Configuring Content Manager OnDemand for Tivoli Storage Manager archive management” on page 92.

Additional optional components are covered in the guide, such as a Tivoli Storage Manager Administration Center, that can assist you in supporting your Tivoli Storage Manager Server.

5.2.2 Configuring Content Manager OnDemand for Tivoli Storage Manager archive management

To enable Content Manager OnDemand to use Tivoli Storage Manager as the archive manager for the system, you must set Content Manager OnDemand options to allow the system to recognize that Tivoli Storage Manager is configured for archive storage.

In a Content Manager OnDemand for Windows system, the Content Manager OnDemand configurator is used to set this parameter. In a Content Manager OnDemand for UNIX or Linux system, the ars.cfg configuration file is updated to specify that Tivoli Storage Manager is used.

In this section, we describe how you can configure Content Manager OnDemand to use Tivoli Storage Manager on both Windows and UNIX or Linux systems.

Content Manager OnDemand for Windows Tivoli Storage Manager configuration

If you are configuring a Content Manager OnDemand for Windows system to use Tivoli Storage Manager for archive storage, the Content Manager OnDemand configurator is used. Either during the creation of the instance or after the instance is created, you can select Tivoli Storage Manager (TSM) as the storage option (Figure 5-2). Click **TSM**, click **TSM Options**, and then enter the path to the Tivoli Storage Manager program files and the path to the Tivoli Storage Manager options file.

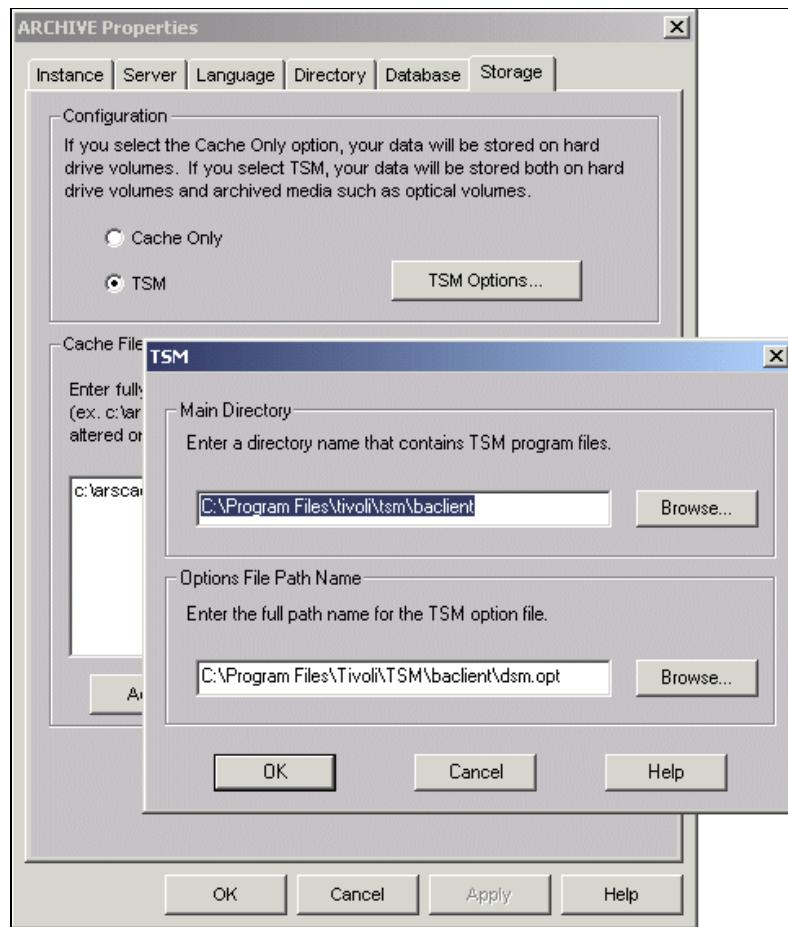


Figure 5-2 Windows configurator

Content Manager OnDemand for UNIX or Linux Tivoli Storage Manager configuration

If you are configuring a Content Manager OnDemand for UNIX system to use Tivoli Storage Manager for archive storage, you must ensure that the ars.cfg file (Figure 5-3 on page 94) is updated to reflect that Tivoli Storage Manager is used as the storage manager.

The file must also include valid paths for the Tivoli Storage Manager options files and all of the Tivoli Storage Manager components that are used. The parameters in the file are used to reference the first Tivoli Storage Manager Server. A single object server can reference multiple Tivoli Storage Manager servers.

```
#####
# Storage Manager Parameters (Library/Object Server) #
#####
#
# Storage Manager for OnDemand to use
#
ARS_STORAGE_MANAGER=TSM

#####
# TSM Parameters (Object Server Only) #
#####
DSMSERV_DIR=/usr/tivoli/tsm/server/bin
DSMSERV_CONFIG=/usr/tivoli/tsm/server/bin/dsmserv.opt
DSM_DIR=/usr/tivoli/tsm/client/api/bin64
DSM_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
DSM_LOG=/tmp
DSMG_DIR=/usr/tivoli/tsm/client/api/bin64
DSMG_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
DSMG_LOG=/tmp
DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
DSMI_LOG=/tmp
```

Figure 5-3 ARS.CFG file for Tivoli Storage Manager configuration

Note: For the Tivoli Storage Manager client that is used by Content Manager OnDemand, set COMPRESSION NO in the Tivoli Storage Manager client option file (dsm.opt for Windows or dsm.sys for UNIX). Content Manager OnDemand objects are compressed before they are sent to Tivoli Storage Manager for archival; therefore, compression by Tivoli Storage Manager is not required.

5.2.3 Content Manager OnDemand storage management

The storage management criteria that you specify for the Content Manager OnDemand library server determines where and when Content Manager OnDemand stores reports and how those reports are maintained.

Figure 5-4 illustrates Content Manager OnDemand storage object relationships. When a report is loaded into Content Manager OnDemand, it is assigned to an application group. The application group is associated with a storage set. The storage set contains one or more storage nodes that can be used by several application groups that have the same archive storage requirements.

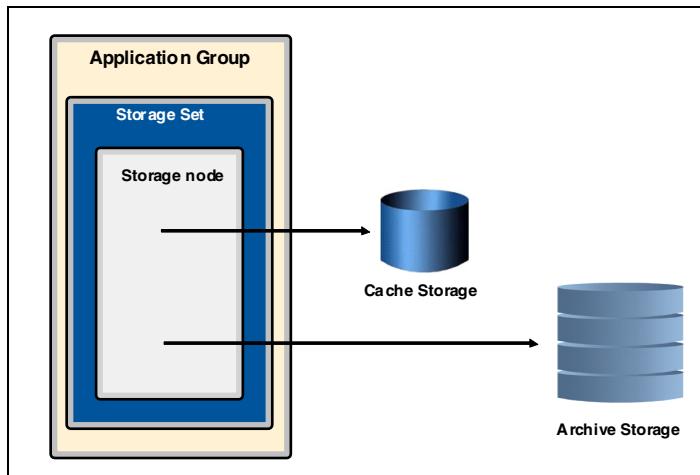


Figure 5-4 Content Manager OnDemand storage objects

For example, a storage set can be used to maintain data from different application groups that must retain documents for the same length of time and require the data to be kept on the same type of media. Different storage sets can be created to handle different data retention requirements. One storage set can be set up to maintain data on cache-only magnetic storage. Another storage set can be set up to point to a Tivoli Storage Manager client node that stores a copy of the report in archive storage.

If Tivoli Storage Manager is used as the ASM, the same storage management criteria must be specified for both Content Manager OnDemand and Tivoli Storage Manager. That is, the *Life of Data and Indexes* in Content Manager OnDemand and the *retention period* in Tivoli Storage Manager must have the same value.

Note: In Content Manager OnDemand, the date that is used to determine the *Life of Data and Indexes* is the date field index value, which is taken from the report that is being loaded and defined as the *segment date*.

For Tivoli Storage Manager:

- ▶ If RETINIT=CREATION, the date that is used for the *retention period* in Tivoli Storage Manager is the date that the report is first migrated to Tivoli Storage Manager.
- ▶ If you are using RETINIT=EVENT, the **RETMIN** parameter specifies the minimum number of days that Tivoli Storage Manager stores an object.

The expiration type determines how the data is expired:

- ▶ If the expiration type value for the application group is *load*, a command is issued from Content Manager OnDemand to Tivoli Storage Manager to delete data when the data is expired from Content Manager OnDemand.
- ▶ If the expiration type is *segment* or *document*, a delete command is not issued from Content Manager OnDemand to Tivoli Storage Manager when Content Manager OnDemand expires the data and the data remains in Tivoli Storage Manager until the Tivoli Storage Manager retention period expires. This data is not accessible from Content Manager OnDemand because the indexes are expired in Content Manager OnDemand.

If Tivoli Storage Manager and Content Manager OnDemand are configured to use event-based archiving, Content Manager OnDemand sends a delete command to Tivoli Storage Manager when data is due to be expired. Tivoli Storage Manager then checks to see whether the data is allowed to be deleted based on its **RETMIN** parameter. The benefit of this approach is that Content Manager OnDemand is in total control of when deletions can take place instead of Tivoli Storage Manager deleting data independently.

5.2.4 Storage set definition

A *storage set* can contain one or more primary storage nodes. A *primary storage node* is used to manage reports and resources that are stored in an application group. A *storage node* is associated with a specific Content Manager OnDemand object server. When Tivoli Storage Manager is used for archive storage, each storage node that is associated with storage that is managed by Tivoli Storage Manager must be registered as a client node in a Tivoli Storage Manager policy domain. The Tivoli Storage Manager policy domain properties determine the type of storage devices that are used to maintain the archived data and the length of time that the data is maintained.

Content Manager OnDemand systems can be set up to run as cache-only hard disk drive systems with no migration of the data or indexes, or with an archive system that uses Tivoli Storage Manager to maintain and manage the archive of Content Manager OnDemand documents and indexes over a predetermined period.

When Content Manager OnDemand is installed and the system is initialized, a default cache-only storage set is created. Additional cache storage sets can be defined. Storage sets that are associated with Tivoli Storage Manager client nodes that are tied to specific management policies on the Tivoli Storage Manager servers are used for long-term archive storage.

The Content Manager OnDemand administrator defines and maintains storage sets (Figure 5-5). The load type is the storage set parameter that we examine.

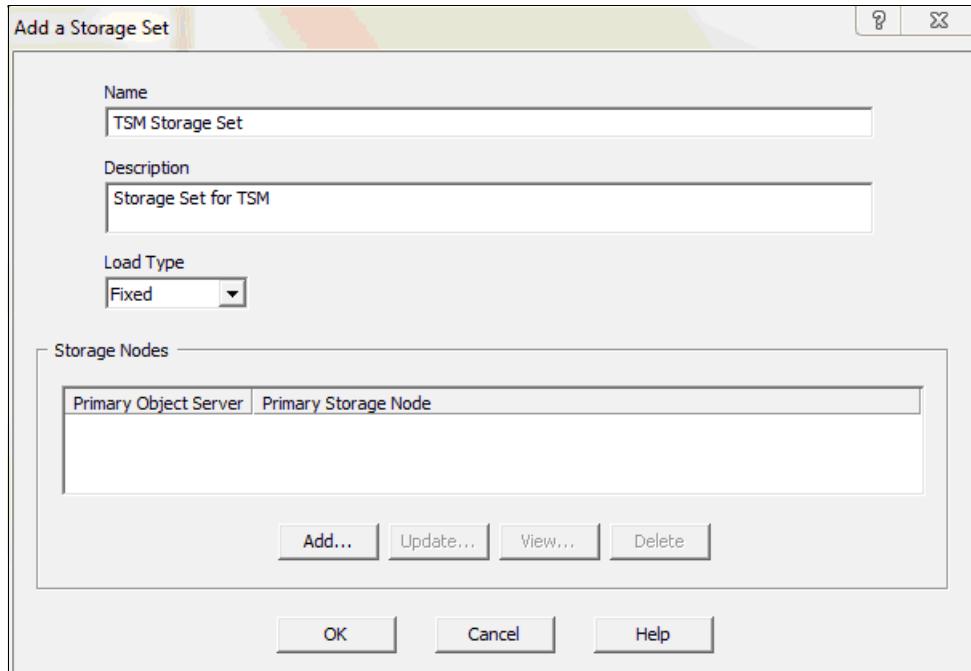


Figure 5-5 Storage set definition

Load Type parameter

The Load Type parameter determines where Content Manager OnDemand stores data. Two values are possible (Figure 5-5):

- ▶ Fixed: Content Manager OnDemand stores data in the primary storage node that has the load data field selected. When Load Type is set to *Fixed*, you must select the load data check box for one primary storage node. Content Manager OnDemand loads data to only one primary storage node regardless of the number of primary nodes that are defined in the storage set.
- ▶ Local: Content Manager OnDemand stores data in a primary storage node on the server on which the data loading program runs. When the Load Type is *Local*, the load data check box must be selected for a primary storage node on each of the object servers that is identified in the storage set. A storage set can contain one or more primary storage nodes that are on one or more object servers.

Next, we examine several parameters on the Add a Primary Node window (Figure 5-6 on page 98).

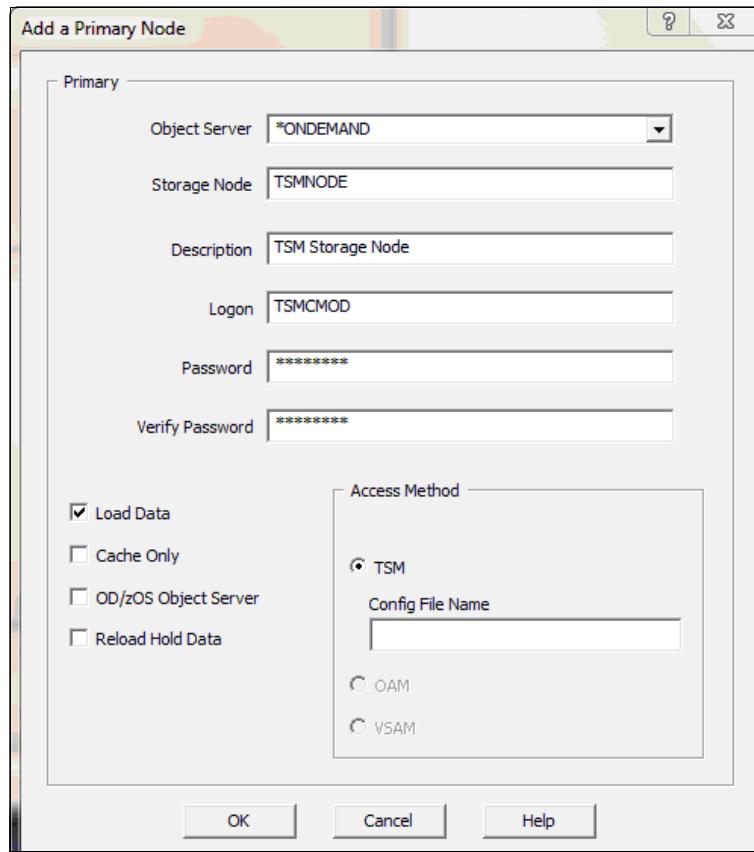


Figure 5-6 Primary storage node definition

Storage node

The Content Manager OnDemand storage node name can be 1 - 60 characters in length and can include embedded blanks. The case can be mixed.

Content Manager OnDemand no longer supports adding auxiliary storage nodes when you create a storage set.

Note: The Content Manager OnDemand storage node name does not tie the storage set to the Tivoli Storage Manager client node. This name is only a label in the Content Manager OnDemand system. The storage node name can be the same as the associated client node name, but they are not required to be the same name.

Logon

If Tivoli Storage Manager is used to maintain archive data, the Logon field is the name of the Tivoli Storage Manager client node. This field is ignored if you are defining a cache-only storage node.

Note: The Logon field must be a valid Tivoli Storage Manager client node name. This client node name is the client node that is defined on the Tivoli Storage Manager system through the wizard or command line. The password that follows the logon must be the same as the password that you created for the client node. Content Manager OnDemand uses the Tivoli Storage Manager application programming interface (API) to connect and log on to the Tivoli Storage Manager server when data is being migrated to the Tivoli Storage Manager client node.

Load Data

The Load Data parameter determines the primary storage node into which Content Manager OnDemand loads data. When the Load Type is Fixed, Load Data must be selected for one primary storage node. When Load Type is Local, Load Data must be selected for one primary node for each object server that is associated with the storage set.

Cache Only

The Cache Only parameter determines whether Content Manager OnDemand uses the archive manager for long-term storage of data.

After you install and configure Tivoli Storage Manager, create a Content Manager OnDemand storage set, and assign it to a Tivoli Storage Manager client node, you are ready to consider how an application group uses the cache storage manager. You must also consider how the ASM stores, maintains, and expires Content Manager OnDemand report data.

Access Method

When you configure Content Manager OnDemand for Multiplatforms Tivoli Storage Manager support, you can specify access to one or more Tivoli Storage Manager servers from a single object server. Only one Tivoli Storage Manager server can be set up to load data at a time by using the Load Data flag. To configure the support for multiple Tivoli Storage Manager servers, you specify the client configuration file under the Config File Name section.

Content Manager OnDemand Object Servers on z/OS

In the Access Method section (Figure 5-7), choose from OAM (object access method) or VSAM (Virtual Storage Access Method) for the access method. OAM is the default access method for the primary storage node. If you choose OAM, specify the collection name. If you choose VSAM, specify the high-level qualifier (HLQ).

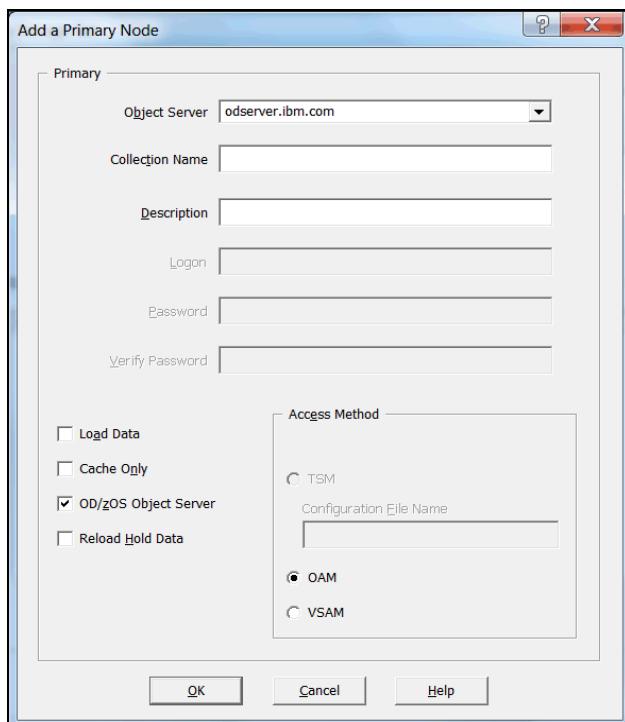


Figure 5-7 Content Manager OnDemand Object Servers on z/OS

Reload Hold Data

You can optionally select the Reload Hold Data check box (Figure 5-8). If it is selected, all documents on hold are reloaded into the storage node after they reach their expiration date.

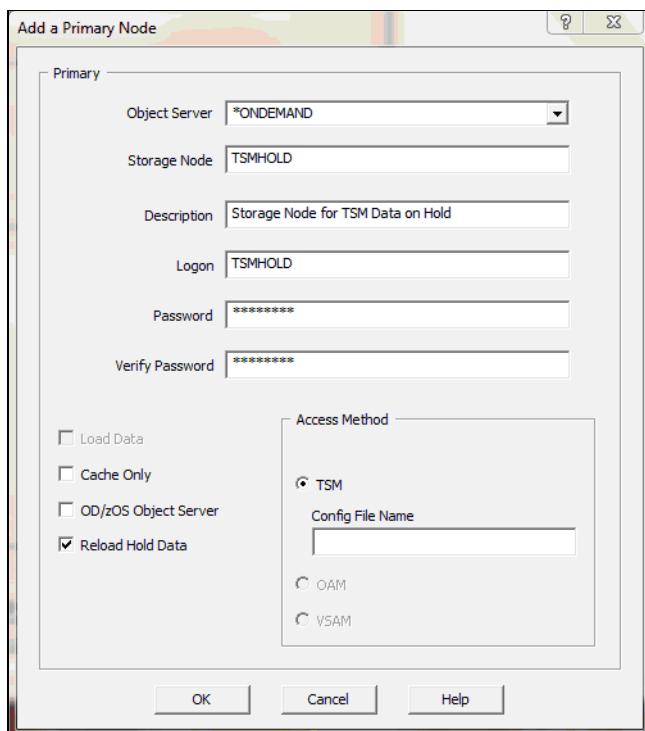


Figure 5-8 Reload Hold Data

For each storage set, you can identify only one storage node as the node where the hold data is reloaded. You can change the Reload Hold Data option when a storage node is updated. This option grants you control of the type of media that reloaded data is placed on that is technically eligible for expiration but is on hold. The location where the held data is stored needs to be managed differently than new data that is being loaded to the system. You do not want to reload Hold Data to a Storage Set/Pool that is defined for 7-year storage.

5.2.5 Application group storage management

The application group storage management settings (Figure 5-9 on page 101) determine how long report data and indexes are kept in cache storage before they are expired. You must decide how soon data is migrated to the archive storage after data is loaded.

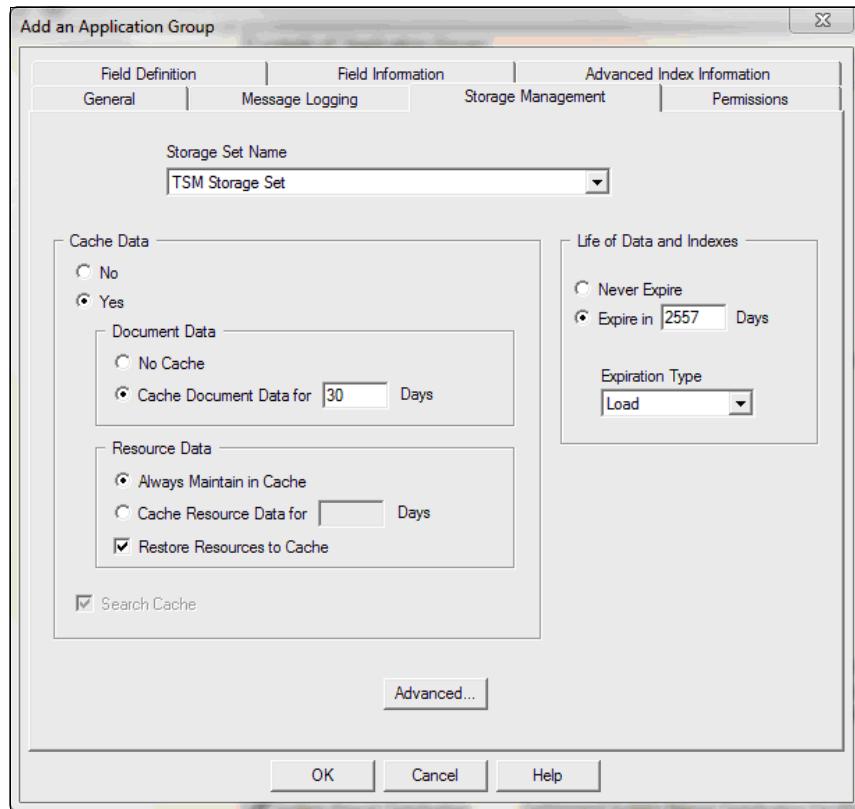


Figure 5-9 Application group storage management

Cache Data

The Cache Data setting determines whether the report data is stored in a hard disk drive cache and, if so, how long it is kept in cache before it is expired. You can also choose whether to search cache when users retrieve documents for viewing. If you choose not to store reports in cache, you must select a storage set that supports archive storage.

Note: Data that is retrieved often needs to generally remain in cache until it is no longer needed by 90% of Content Manager OnDemand users.

Life of Data and Indexes

The Life of Data and Indexes settings determine the length of time that report data, indexes, and resources are maintained in the Content Manager OnDemand system before they are deleted from the application group. The report data, indexes, and resources can be maintained indefinitely if set to never expire, or they might be kept for up to 273 years. After the maintenance threshold is reached, the **arsmaint** command can be used to expire the data from the system.

Expiration Type

The Expiration Type option determines how report data, indexes, and resources are expired. Three expiration types are available:

- ▶ Load: With this expiration type, a single input file (a Load) at a time can be deleted from the application group. The latest date in the input data and the Life of Data and Indexes determine when Content Manager OnDemand deletes the data. Content Manager OnDemand signals to the storage manager that the data might be deleted.

Figure 5-10 shows the error message that displays when you use Enhanced Retention Management and you do not set the expiration type to Load.

Note: Load is the suggested expiration type.

If any application group uses either the Enhanced Retention Management feature or IBM Enterprise Records, this setting is required. You must also use this type if event-based processing is used within Tivoli Storage Manager.

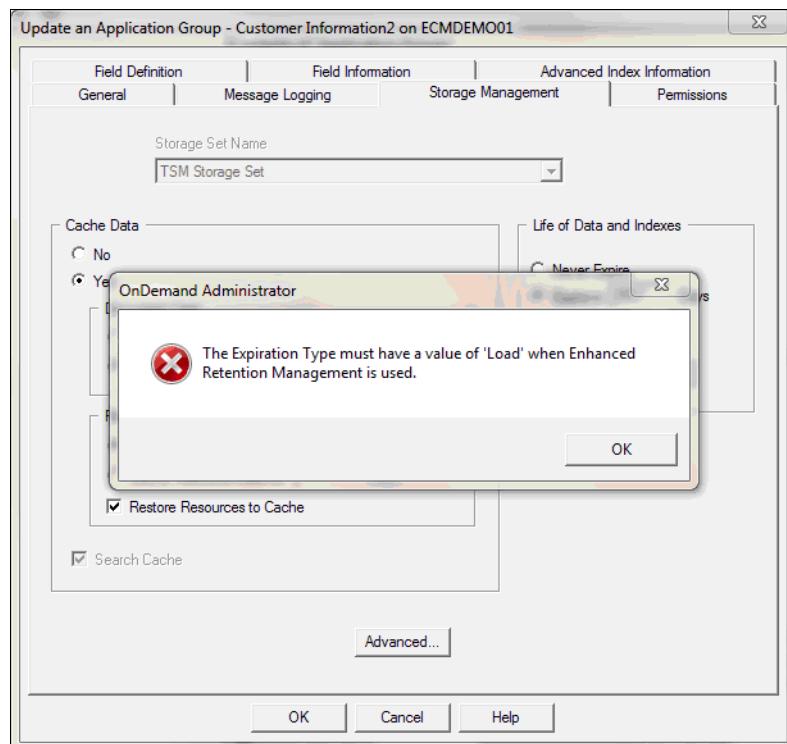


Figure 5-10 Expiration type set incorrectly

- ▶ Segment: With this expiration type, a segment of data at a time is deleted from the application group. The segment must be closed and the expiration date of every record in the segment must be reached. Data that is stored in archive storage is deleted by the storage manager based on the archive expiration date. If a small amount of data is loaded into the application group, and the Maximum Rows value is high, the segment might be open for a long period, and the data is not expired for the period.

- Document: With this expiration type, a document at a time is deleted from the application group. Data that is stored in archive storage is deleted by the storage manager based on the archive expiration date. Storing documents with an expiration type of Document causes the expiration process to search through every document in the segment to determine whether the expiration date was reached, which results in long processing times.

When the **arsmain** expiration process is run, data is deleted only from the application group if the upper threshold for the size of the cache storage is reached. By default, the cache threshold is 80%. A lower threshold can be forced by the expiration command parameters. Unless a reason exists to clear cache, leaving data in cache improves retrieval performance.

5.2.6 Advanced application group storage management

By using the advanced storage management settings (Figure 5-11), you can adjust the size of the load object and determine when report data, indexes, and resources are migrated to archive storage.

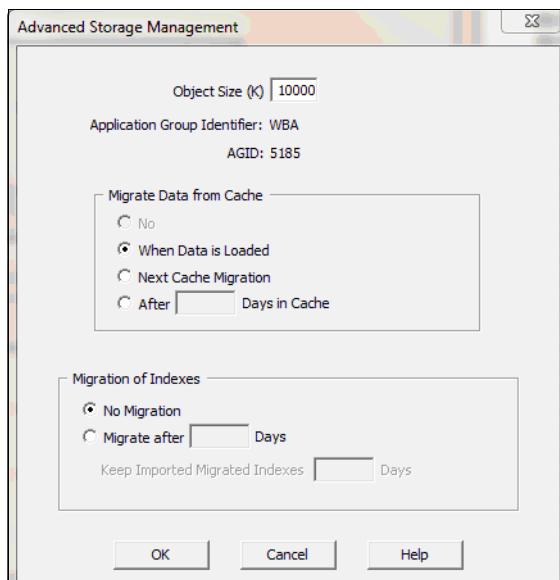


Figure 5-11 Advanced application group storage management

Object Size

The Object Size parameter determines the size of a storage object in kilobytes (KB). Content Manager OnDemand, by default, segments and compresses stored data into 10 MB storage objects. The default of 10 MB is the most commonly used object size value.

Important: Be careful when you change the value for Object Size. Setting the value too small or too large can adversely affect load performance. However, increasing this value might be necessary if you load large files and run out of Object IDs during the loading process.

Note: The object size that is defined here must be equal to or larger than the size of the compressed storage objects that are defined in any application that is assigned to the application group.

Application Group Identifier and the Application Group ID

The Application Group Identifier and the Application Group ID (AGID) are unique identifiers that are used by Content Manager OnDemand to identify the application group in system tables.

Migrate Data from Cache

The Migrate Data from Cache value determines when documents and resources are migrated to archive storage. A storage set that is associated with a Tivoli Storage Manager client node must be selected to enable migration to archive storage.

The following values are valid:

- ▶ No: Data is never migrated from cache. This option is unavailable when a storage set that is associated with a Tivoli Storage Manager client node is selected for the application group.
- ▶ When data is loaded: Data is migrated to archive storage when the data is loaded into the application group.
- ▶ Next cache migration: Data is migrated to archive storage the next time that **ARSMAINT** is run with the **-m** option. The **-m** option indicates that data and resources are copied from cache to archive storage.
- ▶ After __ days in cache: This value specifies the number of days that data remains in cache storage. After the prescribed number of days in cache storage are reached, the data is copied to archive storage the next time that **ARSMAINT** is run with the **-m** option for data migration.

5.2.7 IBM System Storage Archive Manager

Certain regulations require data to be stored in devices that are read only. In the past, physical storage devices, such as tapes and optical disks that are Write Once Read Many (WORM), were used.

WORM disks, such as the NetApp SnapLock or EMC Centera, can be used to store data in the same manner as WORM tapes or optical platters. IBM System Storage Archive Manager allows critical data to be retained for a mandated period without the possibility of being rewritten or erased.

In this section, we describe System Storage Archive Manager and how Content Manager OnDemand can be configured to use this subsystem to support these WORM disk devices.

Note: Verify support for any particular device on a particular platform through the Tivoli Storage Manager Device support matrix before you plan your implementation.

For more information about the Tivoli Storage Manager support of WORM disk devices, such as NetApp SnapLock, or EMC Centera, see the following IBM Knowledge Center documents:

- ▶ *Tivoli Storage Manager for AIX Administrator's Guide*
- ▶ *Tivoli Storage Manager for Windows Administrator's Guide*

You can obtain these documents from the IBM Tivoli Storage Manager Knowledge Center at the following web address:

<http://www.ibm.com/support/knowledgecenter/SSGSG7/welcome?lang=en>:

IBM System Storage Archive Manager

The IBM System Storage Archive Manager feature is sold as a separately licensed software product that is integrated into Tivoli Storage Manager Server software. It requires a stand-alone Tivoli Storage Manager Extended Edition server to be dedicated for its use. It is accessible solely through the Tivoli Storage Manager API by various content management or archive software applications.

For more information about the IBM System Storage Archive Manager, go to the following website:

<http://www.ibm.com/software/products/en/ibmsyststorarchmana>

IBM System Storage Archive Manager provides support in the following key areas:

- ▶ Data retention protection (DRP): Data is not deleted until the retention criteria for the object is satisfied. This feature affects Content Manager OnDemand on loads, unloads, application group deletes, and the expiration of data.
- ▶ Event-based retention policy: Data is retained based on a time interval after the occurrence of a retention-initiating event. For Content Manager OnDemand, this event is a call to delete the data. A load, an unload, application group delete, or expiration of data triggers the retention event.
- ▶ Deletion hold: Data is not deleted or modified until the deletion hold is released. Content Manager OnDemand does not take advantage of this feature. Content Manager OnDemand uses its own built-in deletion hold mechanism that is called Enhanced Retention Management.
- ▶ New device support: Support is available for all of the devices that Tivoli Storage Manager Extended Edition supports.

Content Manager OnDemand operation with the Tivoli Storage Manager server API

With the new event-based retention policy, the object expiration can now be event-based instead of creation-based. A new option is available in the archive copygroup definition that is called **RETINIT**. It determines the time when the retention time that is specified by the **RETV** attribute is initiated. Two values are possible:

- ▶ Creation: This value specifies that the retention time that is specified by the **RETV** attribute is initiated at the time that an archive copy is stored on the Tivoli Storage Manager server.
- ▶ Event: This value specifies that the retention time that is specified in the **RETV** parameter is initiated at the time that a client application notifies the server of a retention-initiating event for the archive copy. If you specify **RETINIT=EVENT**, you cannot also specify **RETV=NOLIMIT**.

We compare the behavior of Tivoli Storage Manager when Content Manager OnDemand data is deleted with the previously listed two options together with the setting of data protection.

Table 5-1 shows the action by Tivoli Storage Manager when a Content Manager OnDemand object is deleted, unloaded, or during the deletion of an application group when data protection is turned OFF.

Table 5-1 Comparison of expiration methods with data protection OFF

Tivoli Storage Manager RETINIT	Content Manager OnDemand action: Unload	Content Manager OnDemand action: Delete application group
Creation	The Delete Object command is issued through the Tivoli Storage Manager API. Objects are deleted during the next Tivoli Storage Manager expiration.	The Delete Filespace command is issued. Objects are immediately deleted with the file space.
Event	Content Manager OnDemand issues an event trigger command through the Tivoli Storage Manager API. The status of the objects that are affected is changed from PENDING to STARTED and is expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.	The Delete Filespace command is issued. Objects are immediately deleted with the file space.

Table 5-2 shows the action by Tivoli Storage Manager when data protection is turned ON.

Table 5-2 Comparison of expiration methods with data protection ON

Tivoli Storage Manager RETINIT	Content Manager OnDemand action: Unload	Content Manager OnDemand action: Delete application group
Creation	Content Manager OnDemand issues no commands to Tivoli Storage Manager. The objects are effectively orphaned by Content Manager OnDemand and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.	Content Manager OnDemand issues no commands to Tivoli Storage Manager. The objects are effectively orphaned by Content Manager OnDemand and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.
Event	Content Manager OnDemand issues an event trigger command through the Tivoli Storage Manager API. The status of the objects that are affected are changed from PENDING to STARTED and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.	The Delete Filespace command cannot be used with DRP ON, so the operation is treated the same as though a delete were indicated and the status of all of the affected objects is changed from PENDING to STARTED. They are expired by Tivoli Storage Manager based on their retention parameters. This action unfortunately leaves the file space entries in Tivoli Storage Manager. These entries can be manually deleted after the file space is empty even with DRP ON.

Content Manager OnDemand version 9 setup recommendations

The following recommendations are applicable to Content Manager OnDemand V9.0 and later:

- ▶ Application groups need to be set up to expire by load; then, you can use the Enhanced Retention Manager document hold feature.
- ▶ Tivoli Storage Manager archive copy groups need to be defined to be event-based by setting the **RETMIN** and **RETVER** parameters. The **RETMIN** parameter needs to be set to the minimum number of days that a document must be retained. For a legal 7-year document, this setting must be 2557. For others, where you want Content Manager OnDemand to be 100% in control and able to delete documents at anytime, set RETMIN=0. Content Manager OnDemand then issues a delete based on its Life of Data and Indexes or when an administrator performs an unload command.
- ▶ Tivoli Storage Manager Inventory expiration must be run regularly to ensure that expired data is cleaned up.

5.2.8 The **arsmaint** command

We referenced the Content Manager OnDemand **arsmaint** command many times in previous sections, but we now look closer at this command. The **arsmaint** program maintains application group data that is stored in the Content Manager OnDemand database and in cache storage. It maintains the system by using the storage management values that are specified for application groups. It is typically run on a regular schedule to migrate documents from cache storage to archive storage, migrate index data to archive storage, and delete documents from cache storage and index data from the Content Manager OnDemand database.

The **arsmaint** command uses the application group expiration type to determine how to delete index data from an application group. This command can expire a table of application group data at a time (segment expiration type), an input file of data at a time (load expiration type), or individual documents (document expiration type).

Note: When cache data is expired, by default the data is not expired until the cache storage file system exceeds 80% of capacity. Keeping data in cache as long as possible improves retrieval and viewing performance. You can force the expiration of cache data before the cache is 80% full by using the minimum and maximum parameters to override the percentage full default.

For a detailed explanation of the **arsmaint** command and its associated parameters, with all of the other Content Manager OnDemand commands, see *IBM Content Manager OnDemand for Multiplatforms, V9.5, Administration Guide*, SC19-3352.

5.3 Object access method for z/OS

In this section, we provide an introduction to object access method (OAM) and show its relationship with Content Manager OnDemand in a z/OS environment.

For more information about setting up OAM, see the following documentation:

- ▶ *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426
- ▶ Chapter 3, “OAM and System Management Subsystem customization”, in *Image and Workflow Library: Content Manager for ImagePlus on OS/390 Implementation and EIP*, SG24-4055

OAM is a hierarchical data management system (disk → optical → tape) that is used for archive storage.

Note: When you use OAM as the storage manager, Content Manager OnDemand can retrieve the stored object directly from disk, optical drive, or tape.

OAM uses the OSREQ macro interface and uses DB2 both for its internal (indexing) tables and for online storage objects. OAM is the DFSMSdfp component that manages a class of data, which is called *objects*, in a z/OS environment. *Objects* are bit strings that are handled as one large byte string rather than processing them as records, as is done with datasets. The content of this byte string is not known to OAM. No restrictions exist on the data type of this object; it can be an image, compressed data, or coded data.

How to handle this data is left up to the application. OAM handles an unlimited number of objects, which can be stored on magnetic disk, magnetic tape, or optical storage. Objects are different from datasets, which are handled by existing access methods. The following characteristics distinguish objects from traditional datasets:

- ▶ Lack of record orientation: No concept of individual records within an object exists.
- ▶ Broad range of size: An object might contain less than 1 KB or up to 256 MB of data.
- ▶ Volume: Objects are much smaller than datasets; however, they can use much more external storage, depending on the type of application that creates them, such as image applications.
- ▶ Access time requirements: Reference patterns for objects change over time, allowing less critical objects to be placed on lower-cost, slower devices or media.

5.3.1 OAM components and SMS terminology

In this section, we describe the three components of OAM and also OAM terminology.

OAM components

OAM functions are performed by three components:

- ▶ Object Storage and Retrieval (OSR) component

This component provides an API for OAM. All OAM API functions are requested through the OSREQ assembler macro. Applications use this interface to store, retrieve, query, and delete objects, and to change information about objects. OSR stores the objects in the storage hierarchy and maintains the information about these objects in DB2 databases. OSR functions that start through the API require the OAM Thread Isolation Support (OTIS) application for administrative processing.

- ▶ Library Control System (LCS) component

This component writes and reads objects on tape and optical disk storage. It also manipulates the volumes on which the objects are stored. The LCS component controls the usage of optical hardware resources that are attached to the system.

- ▶ OAM Storage Management Component (OSMC)

This component determines where to store objects in the OAM storage hierarchy. It manages object movement within the object storage hierarchy and manages expiration attributes that are based on the installation storage management policy that is defined through the storage management subsystem (SMS). OSMC also creates the requested backup copies of the objects and provides object and volume recovery functions.

SMS terminology

To provide a better understanding of OAM, we explain SMS terms in the following sections.

SMS storage class

A *storage class* is a collection of performance goals and availability and accessibility requirements that are defined to SMS. It is used to select a device to meet those goals and requirements.

Usually, three storage classes are set up for OAM where the storage administrator sets the names of the storage classes based on the naming convention in the enterprise. The three OAM storage classes to set up are listed:

- ▶ OAMDASD: Objects are stored in a DB2 table on fast magnetic disk.
- ▶ OAMTAPE: Objects are stored on magnetic tape, including tape robots.
- ▶ OAMOPTIC: Objects are stored on a 3995 optical device.

Note: The Content Manager OnDemand cache storage on a hierarchical file system (HFS) is not part of these SMS constructs.

SMS storage group

An SMS *storage group* is a collection of storage volumes and attributes that are defined by the installation. Storage groups, with storage classes, help reduce the requirement for users to understand the physical characteristics of the storage devices that contain their data.

In an OAM environment, object storage groups allow the storage administrator to define an object storage hierarchy. The object storage hierarchy classifies storage areas according to location and, therefore, according to retrieval response time. Each object storage hierarchy must contain an object directory, containing control information about each object.

Additionally, the hierarchy can have the following items:

- ▶ DB2 object storage tables on a hard disk drive
- ▶ Optical volumes that are associated with optical libraries (real or pseudo), and stand-alone or operator-accessible optical disk drives
- ▶ Tape volumes that are associated with tape libraries or stand-alone tape drives

SMS management class

Management classes define the space and availability requirements for datasets. Class attributes control backup, migration, retention of data, and release of unused space. OSMC uses information from the management classes to determine the automatic management processes that need to be performed on corresponding OAM objects.

Automated Class Selection routine

Automated Class Selection (ACS) routines are used to assign class and storage group definitions to datasets and objects. ACS routines are written in the ACS language, which is a high-level programming language that is similar to the language that is used for the construction of TSO CLISTS. The ACS translator is used to convert the routines to object form so that they can be stored in the SMS configuration.

OAM collection

A *collection* is a group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. A collection is used to catalog many objects, which, if cataloged separately, can require a large catalog. Every object must be assigned to a collection. Object names within a collection must be unique; however, the same object name can be used in multiple collections. Each collection belongs to only one object storage group. Each storage group can contain from one to many collections.

Important: A collection is the only interface that is used by the administrator to determine how to store objects in OAM. It is used when you create a storage set.

5.3.2 OAM configuration recommendations

This section provides a list of recommendations for you to review and consider when you configure OAM for Content Manager OnDemand. They are classified in the following categories:

- ▶ General
- ▶ DB2
- ▶ Devices
- ▶ Tapes
- ▶ Maximum Object Size (MOS)
- ▶ Optical platters
- ▶ ARS.CFG setting

General

Consider the following general recommendations when you work with OAM for Content Manager OnDemand:

- ▶ Define a user catalog exclusively for collection names.
- ▶ Cache the user catalog in the virtual lookaside facility (VLF).
- ▶ Migrate objects to optical or tape (OSMC) during non-peak hours.
- ▶ Spread OAM collections over multiple DB2/disk/channels.
- ▶ Spread out the application groups to different collection names:
OAM collections → storage groups → DB2 database
- ▶ Group your applications based on retrieval expectations:
 - Collect small, frequently used applications together.
 - Isolate your important applications so that the other applications do not get in the way.

DB2

Consider the following list of recommendations that relate to DB2:

- ▶ Ensure that enough DB2 connections are available to support the OAM requests.
- ▶ Run the **REORG**, **RUNSTATS**, and **REBIND** commands, as appropriate.
- ▶ Partition OAM table spaces larger than 2 GB.

Devices

Consider the following recommendations that relate to devices:

- ▶ Determine and set the Initial Access Response Seconds (IARS) option.
- ▶ Assign objects to storage classes that have an adequate IARS that is defined and to management classes that do not cause a transition to a slower storage class until the frequency of retrieving the objects is reasonably low.
- ▶ Determine whether to place the object on disk or removable (optical or tape) media.
- ▶ If the IARS opts for REMOVABLE media, determine whether to place the object on optical or tape.
- ▶ Verify that the required Sustained Data Rate is achieved based on the selected object placement.

Tapes

Consider the following recommendations that relate to tapes:

- ▶ Modify (CBROAMxx) **MAXTAPERETRIEVETASKS** and **MAXTAPESTORETASKS** (if you are using tape retrieves because the default=1). Both of these parameters are configured at the global level and at the storage group level.
- ▶ The global level **MAXTAPERETRIEVETASKS** (tasks) is defined by **SETOAM**. **SETOAM** specifies the total number of concurrent tape retrievals that are possible at a time. (It controls the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading object data from tape.) This number must be less than or equal to the number of tape drives on the system. Do not specify a number that is greater than the number of tape drives that are available to OAM for the **MAXTAPESTORETASKS** or the **MAXTAPERETRIEVETASKS** subparameters because a system can go into allocation recovery and attempt to allocate tape drives after all of the tape drives are in use, causing system problems.
- ▶ The storage group level **MAXTAPERETRIEVETASKS** (tasks) specifies the maximum concurrent tape retrievals that are possible for a specific storage group. If **MAXTAPERETRIEVETASKS** is

not set, the default for each storage group is 1. This value must be set for each storage group that requires a value larger than the default of 1. For a single storage group, you must set this parameter if you must retrieve documents from two or more tapes concurrently.

- ▶ Set the OAM cataloged procedure parameter **MAXS** (the number of storage groups that the storage management cycle processes concurrently) to an appropriate value. If **MAXS** increases above 10, the effectiveness of concurrency diminishes and OAM processing can be severely constrained or unsuccessful. If concurrent processing includes OBJECT storage groups that write to tape volumes, you must specify the correct corresponding (global level) **MAXTAPERETRIEVETASKS** and **MAXTAPESTORETASKS** values on the **SETOAM** statement.
- ▶ If you are using optical platters or tapes, set the tape **DEMOUNTWAITTIME** time parameter appropriately. This parameter determines how long OAM leaves a tape volume mounted in anticipation of another retrieval request from that device.

Maximum Object Size (MOS)

OAM now supports storing object sizes up to 256 MB. Authorized program analysis report (APAR) OA03623 lists the program temporary fix (PTF) that is available for each release of z/OS. To enable support for object sizes up to 256 MB, you must specify the maximum object size by adding **MOS =256** to the OAM subsystem definition parameter **INITPARM** in the **SYS1.PARMLIB(IEFSSNxx)** member.

The Maximum Object Size (MOS) can be viewed by running the following command:

D SMS,OAM

The results of that command are shown:

```
OAM1 Parm: TIME=GMT MSG=EM UPD=N QB=Y  
MOS= 256 OTIS=N LOB=N DP=N
```

If the MOS is too small, Content Manager OnDemand returns an error similar to “OAM Store Failed with an OSREQ RC=8 and Reason=2402020”. You must increase the MOS size. For more information, review the document at this website:

<http://www.ibm.com/support/docview.wss?uid=swg21408525>

Optical platters

When you work with optical platters, check and adjust the values for the following parameters in **SYS1.PARMLIB(CBROAMxx)**:

- ▶ **MOUNTWAITTIME**: Specifies the amount of time (in minutes) that can pass while a volume waits to be mounted on an operator-accessible drive within an optical library. After this time expires, message CBR4426D is issued to allow the operator to try again or to cancel the volume mount request. This value can be any numeric value 1 - 9999. If the operator retries the mount request, the value that is specified in the **MOUNTWAITTIME** parameter is used for the retry. The default value of this parameter is 5 minutes.
- ▶ **OPTICALDISPATCHERDELAY**: Specifies the number of seconds that the OAM optical dispatcher delays the processing of certain requests to minimize the flipping of optical disk cartridges in an automated optical storage library that expects that another read request for the currently mounted optical disk volume will arrive within this delay interval.

The OAM optical dispatcher delays processing of a unit of work for a specific period, when *all* of the following conditions are true:

- A read request for an object on a currently mounted optical disk volume was completed.

- No request exists for the currently mounted optical disk volume that is waiting to be processed on the OAM optical dispatcher queue.
- The OAM optical dispatcher found a read request for another optical disk volume (either the opposite side of the currently mounted volume or an unmounted optical disk volume) and is about to dispatch this unit of work.
- A nonzero optical dispatcher delay value is specified with the **OPTICALDISPATCHERDELAY** keyword on the **SETOPT** statement in the CBROAMxx PARMLIB member.

If another read request for the currently mounted optical disk volume arrives within the delay interval, that unit of work is dispatched immediately upon arrival. If no read request for the currently mounted optical disk volume arrives within the delay interval, another request for a different optical disk volume (either the opposite side of the currently mounted optical disk volume or an unmounted optical disk volume) is dispatched. Valid values for seconds are decimal numbers 1 - 60. If usage of this parameter is necessary, use of a low value, 1 - 5, is suggested.

ARS.CFG setting

If you are configuring a Content Manager OnDemand for z/OS system to use OAM for archive storage, you must ensure that the ars.cfg file is updated to reflect that OAM is used as the storage manager. Example 5-1 shows an example of the settings to configure for OAM archive storage enablement for Content Manager OnDemand.

Example 5-1 ars.cfg OAM parameter setting

```
#####
# OAM Parameters (Object Server Only) #
#####
#
# Number of OAM SubServers.
# 0          - OAM is not used
# Otherwise - The number of OAM SubServers to handle
# connections to OAM
#
ARS_NUM_OAMSRVR=20
ARS_OAM_DB2SSID=DB2T
ARS_OAM_PLAN=CBRIDBS
```

5.3.3 Defining a storage set

When the Content Manager OnDemand administrator defines a new storage set, the Add a Storage Set window opens, as shown in Figure 5-12 on page 114.

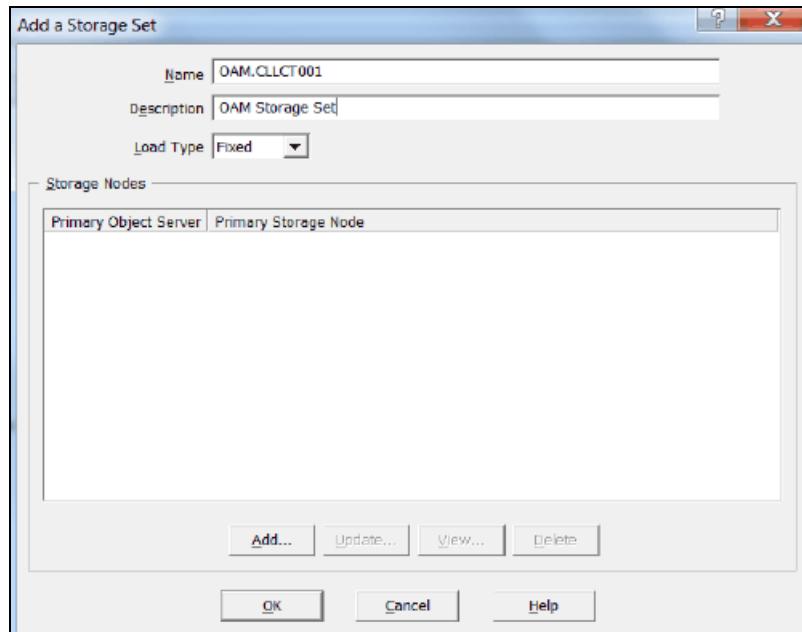


Figure 5-12 Adding an OAM storage set

The administrator must define values for the following fields to add a storage set:

- ▶ Name: The name of the storage set.
- ▶ Description: The storage set description, up to 120 characters.
- ▶ Load Type: Where Content Manager OnDemand stores data. Two choices are available:
 - Fixed: Content Manager OnDemand stores data in the primary storage node that has the load data field selected. When you set load type to Fixed, you must select the **Load Data** check box for one primary storage node. A storage set can contain one or more primary storage nodes. Several different collection names can be used. Content Manager OnDemand loads data in one primary storage node regardless of the number of primary nodes in the storage set.
 - Local: Content Manager OnDemand stores data in a primary node on the server on which the data loading program runs. This load type applies to z/OS.

Then, the administrator clicks **Add** to add a primary storage node to this storage set. The Add a Primary Node window opens, as shown in Figure 5-13 on page 115.

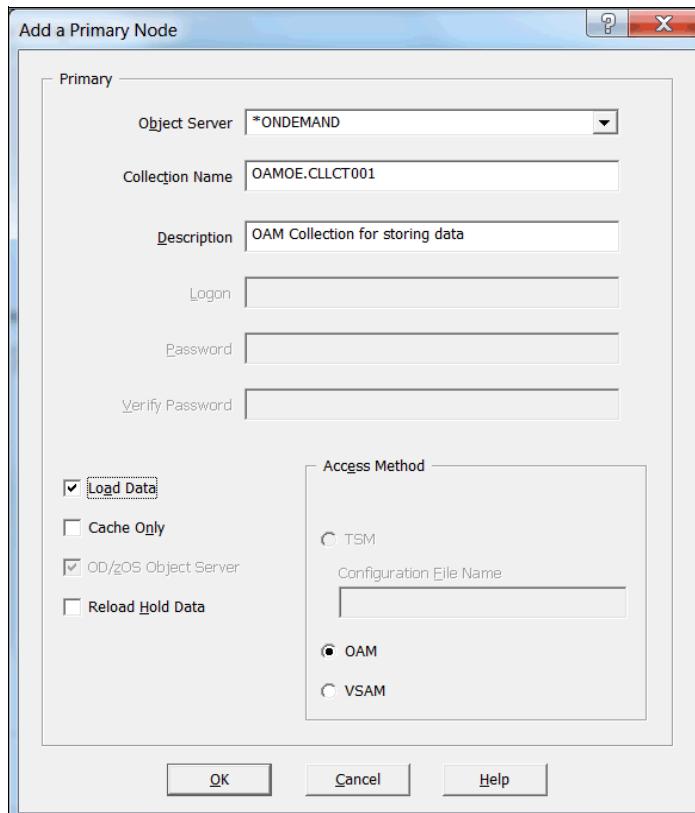


Figure 5-13 Adding a primary storage node to the storage set

The object server is the TCP/IP host name alias, fully qualified host name, or IP address of the server on which the storage node exists. Select a name from the list or enter the name of a Content Manager OnDemand *object server*. Select ***Content Manager OnDemand** if the storage node is on the Content Manager OnDemand *library server*.

The *load data* check box indicates that the data is loaded to this collection. You must select the **OAM** check box. The Logon, Password, and Verify Password fields are used only when Tivoli Storage Manager is selected for the access method.

A one-to-one relationship exists between a collection and a storage set. You can add more primary storage nodes to one storage set, but only one primary storage node can be active at a time.

Figure 5-14 on page 116 shows the relationship between the creation of storage sets and OAM.

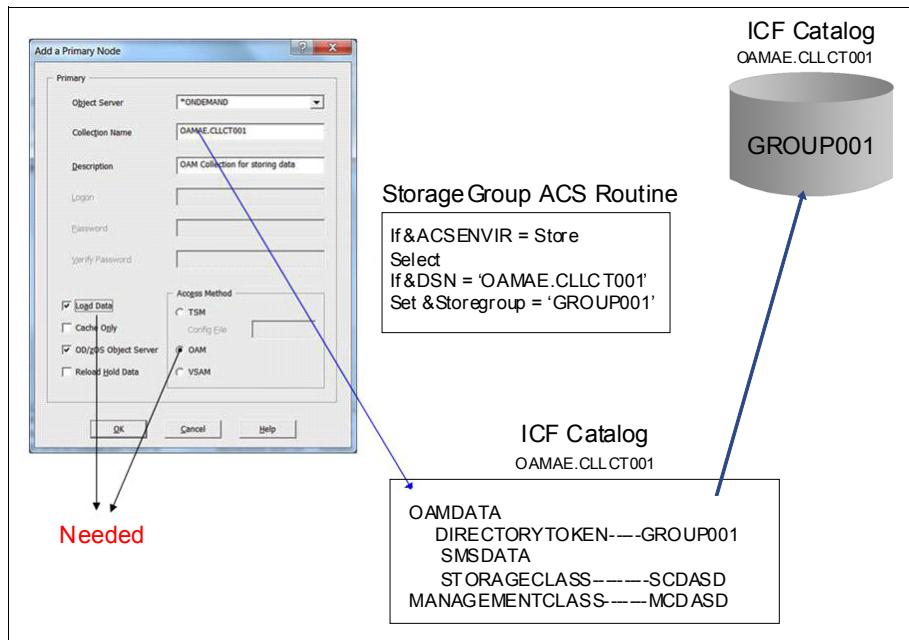


Figure 5-14 Relationship between OAM and Content Manager OnDemand

Object naming conventions

The *object name* identifies the object within a collection. The object name is unique within a collection and it is provided by the Content Manager OnDemand application. Currently, no installation exits allow any customization of these names. The object name is composed of the application group name and the load identifier within the application group portion of the load ID. The load identifier within the application group is composed of a numeric sequence number followed by a character string, such as FAAA. This string is then converted into two qualifiers of the object name:

- ▶ L indicates that the object contains document data.
- ▶ R indicates that the object contains resource data.

The application group name is added, and an object name looks like the following syntax:

A BDA.L1.FAAA

The maximum size of an object is specified through the Content Manager OnDemand Administrator Client when you define an application group. The default value is 10 MB. Currently, the maximum size for an OAM object is 256 MB. The Content Manager OnDemand administrator must be careful not to specify a value that exceeds this limit.

Important: In the current implementation, Content Manager OnDemand is not aware that an object was deleted by OAM based on the management class criteria that are set by the Storage Management component. A user can search for data that is no longer available. No synchronization occurs between OAM object expiration and index expiration. Ensure that you define the index expiration correctly when you define the application group.

Figure 5-15 on page 117 shows the window in which you can set up the index expiration for Storage Management when you define or update an application group.

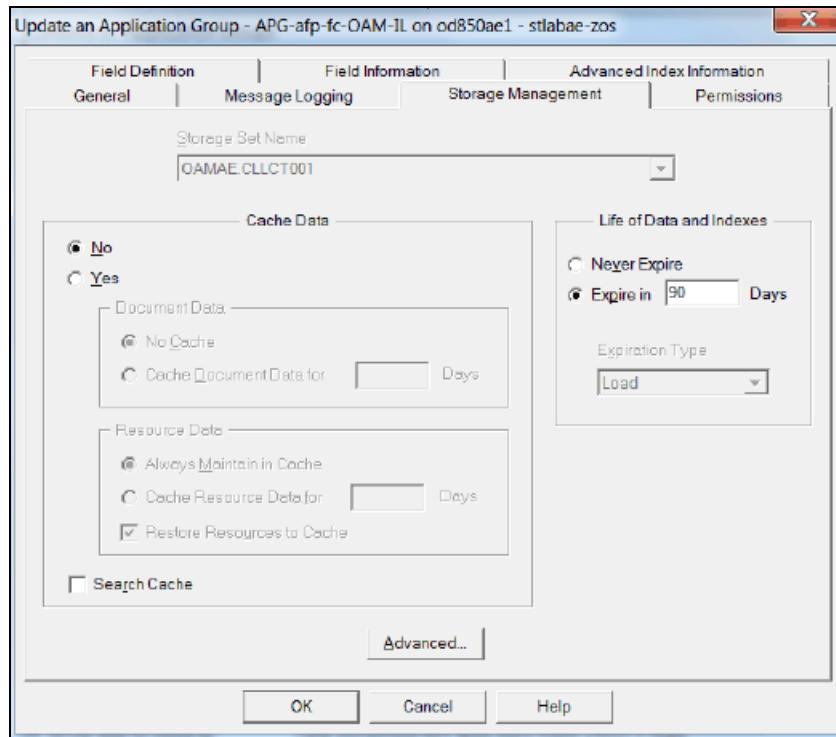


Figure 5-15 Defining index expiration in Content Manager OnDemand

Tip: Content Manager OnDemand and OAM can run in different DB2 subsystems (different DB2 subsystem identifiers (SSIDs)).

5.3.4 Storing data in Virtual Storage Access Method datasets

Another way to store data on the z/OS system is through Virtual Storage Access Method (VSAM). Content Manager OnDemand can create objects that are stored in VSAM datasets. All storage management issues for VSAM datasets, such as allocation, backup, and migration, apply for these object datasets.

To create a storage set that stores objects in VSAM datasets, the Content Manager OnDemand administrator must provide the first-level qualifier for the defined cluster statement. In the example that is shown in Figure 5-16 on page 118, VSAMTST is the high (first) level qualifier.

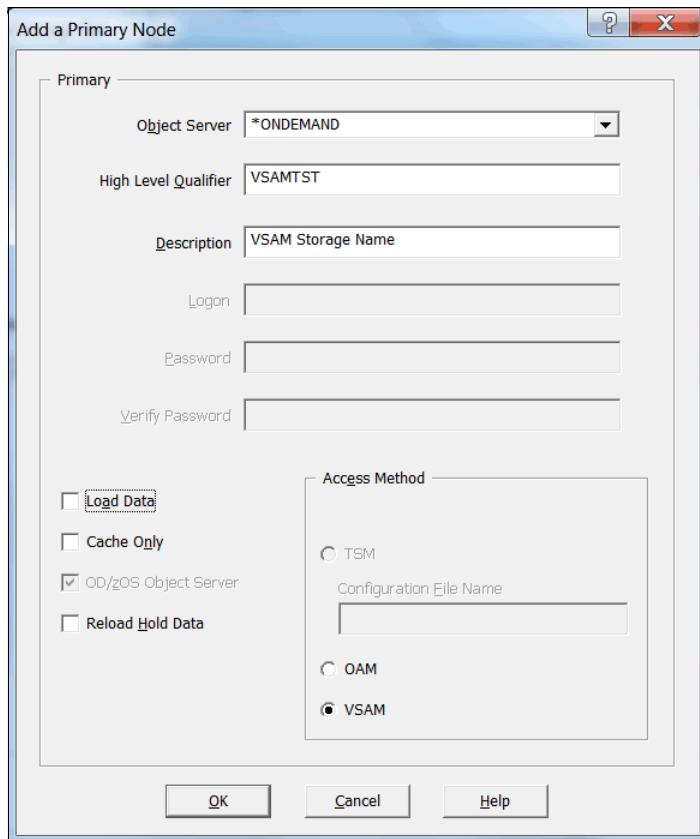


Figure 5-16 Defining a storage set for VSAM

Based on these parameters, Content Manager OnDemand creates VSAM datasets during the **arsload** program. A catalog entry is created, as shown in Example 5-2.

Example 5-2 VSAM dataset name

VSAMTST.FAA.L1.FAAA

This catalog entry is created automatically by the Content Manager OnDemand system. The only part that you can create for yourself is the first-level qualifier. The space allocation during the Define Cluster is performed by the Content Manager OnDemand code, as well. The default object size that is set when you define the application group influences the number of bytes for the primary allocation and the secondary allocation. The number of bytes is divided by 16 for the primary allocation. Every time that an **arsload** command runs with this storage set, this amount of data is allocated even if the objects are much smaller.

Every load creates two VSAM datasets: one VSAM dataset for the data, and one VSAM dataset for the index. Every Define Cluster of a VSAM dataset is a catalog entry. If you have several million loads with this storage set, your catalog grows large.

You can browse the VSAM dataset, but if the compression is on, you cannot see much. For test purposes, compression can be switched off and then the content of the VSAM dataset is viewable. Compression can be switched off on the load information in the application window.

If you store AFP data to VSAM, the resources are stored in a different VSAM dataset.

5.4 Archive Storage Manager for IBM i

The Disk Storage Manager of Content Manager OnDemand for i maintains a copy of documents on disk. Disk Storage Manager migrates documents from cache to the Archive Storage Manager (ASM). ASM then migrates documents to archive media.

ASM maintains one or more copies of documents on archive media, such as disk pool, optical (physical or virtual), or tape. The Content Manager OnDemand administrator decides the type of archive media that is required, configures the storage devices on the systems, and defines the storage devices to ASM. To store application group data on archive media, the application group must be assigned to a storage set that is managed by ASM.

When an application group is created, the Content Manager OnDemand administrator specifies how long the documents must be maintained on the system and whether the index data needs to be migrated from the database to archive media. Content Manager OnDemand system management programs use this information to migrate documents from cache to ASM, delete documents from cache, migrate index data from the database to archive media, and delete index data from the database. Content Manager OnDemand can then reclaim the space that was used by the migrated and expired data.

Disk Storage Manager expires data based on the Life of Data and Indexes value. You can access this setting by clicking **Application Group → Storage Management**. ASM deletes data from the archive media when the data reaches its storage expiration date. The Content Manager OnDemand administrator defines management information to the ASM for the Content Manager OnDemand data that is managed. This management information includes storage volumes that can contain Content Manager OnDemand data, the number of copies of a report to maintain, and the amount of time to keep data in the archive management system.

5.4.1 Migration policy

Migration policies and storage sets must be defined before you can define reports to Content Manager OnDemand or load data into the system. Migration policies contain migration and storage media characteristics for data that is archived by using Content Manager OnDemand. The information is used by ASM to determine whether and when to move archived data through a hierarchy of storage media, such as disk, optical, or tape. Each step in the movement of data through this storage hierarchy is referred to as a *migration policy storage level*. Each migration policy must contain at least one storage level. Additional levels might be defined to meet your storage and retrieval requirements.

The *Cache Only - Library Server* storage set is no longer created automatically with the installation of Content Manager OnDemand for i. The “Cache Only” storage set is limiting because you cannot add any storage levels to it. A better alternative is to define a disk pool and create a migration policy instead. This approach provides the flexibility of adding additional storage levels to the policy later.

When you create a migration policy, a storage set of the same name is automatically created by Content Manager OnDemand. If you plan to keep all of your archives on disk, the best approach is to create a disk pool and to create a migration policy that specifies “No Maximum” for the duration level. Disk Storage Manager expires data and indexes whenever the number of days is reached in the Life of Data and Indexes setting in the application group, or whenever an expiration level in the migration policy is encountered, whichever comes first. If no expiration level is in the migration policy, data is only expired according to the Life of Data in the application group.

If you plan to add a virtual optical level later to take advantage of improved save and restore times or to use the WORM option, you can initially specify 90 days, for example, for the disk pool level, with no other defined storage levels. When a virtual optical level is added later, the archives are moved from the disk pool level to the virtual optical level. With this technique, you must never add an expiration level after the disk pool level because if that level is encountered, the archives are expired.

In the status report that is created by ASM, you might see messages that indicate that the number of days in the ASP01 disk pool level was exceeded because no level is available after 90 days in this example. You can ignore these messages.

If you choose the default in the application group to migrate data from cache when data is loaded, a copy of the data is archived to the integrated file system (IFS) CACHE directory and to the ASMREQUEST directory. When you run Disk Storage Manager, the data is deleted from cache after the Cache Data for Days duration ends. When you run ASM for the first time after you load data, the data is moved to the first level of the migration policy, ASP01 in our example. If aggregation is used, the data is not migrated until the appropriate object size is aggregated, or until the number of days to aggregate was passed. The data remains in ASP01 until the number of days in the Life of Data and Indexes is reached or an expiration level in the migration policy is encountered, whichever comes first.

Most administrative functions for Content Manager OnDemand for i can be carried out with the Content Manager OnDemand Administrator Client. The objects that are necessary for Content Manager OnDemand archive storage management on IBM i are created by using the Content Manager OnDemand component of the web-based IBM Navigator for i (Figure 5-17).

To create a migration policy for use by the archive storage management process, storage devices must be defined for the types of archive media that are required by the Content Manager OnDemand system. For our scenario, we created a disk pool storage group and an optical storage group.

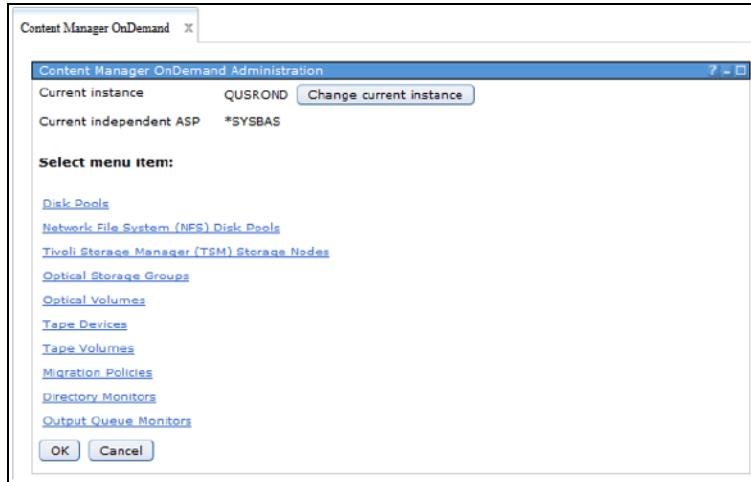


Figure 5-17 IBM Navigator for i

Disk pool storage group

A *disk pool storage group* is used to identify an IBM i auxiliary storage pool that ASM uses as disk storage media when it migrates archived data. Use IBM Navigator for i to add a disk pool storage group (Figure 5-18 on page 121).

Provide the following information for disk pool definition:

- ▶ A pool number that corresponds to an existing auxiliary storage pool
- ▶ A description of the storage group
- ▶ The type of data, which is primary or backup

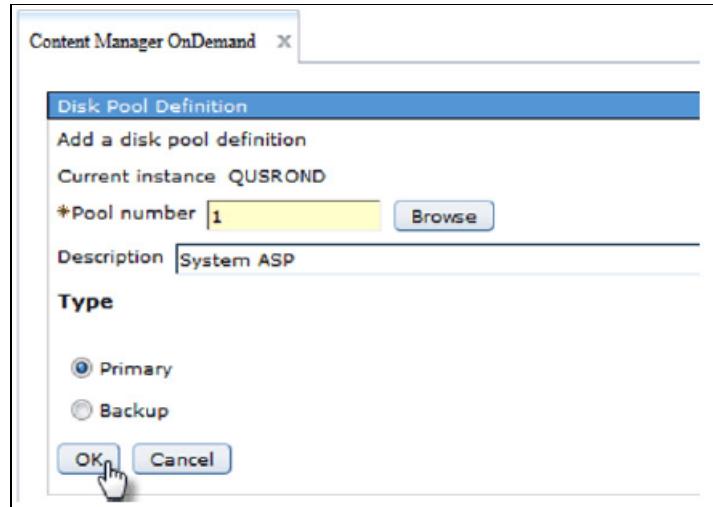


Figure 5-18 Content Manager OnDemand for i disk pool definition

Optical storage group

Optical storage groups are used by Content Manager OnDemand to group sets of optical volumes for the storage of related data. Optical storage groups are used to group physical optical volumes and virtual optical volumes. Each optical storage group must contain only one type (physical or virtual). By using a specific storage group in the migration policy, the administrator can control the sets of reports that are stored on a particular set of optical volumes. Use IBM Navigator for i to define the optical storage group (Figure 5-19).

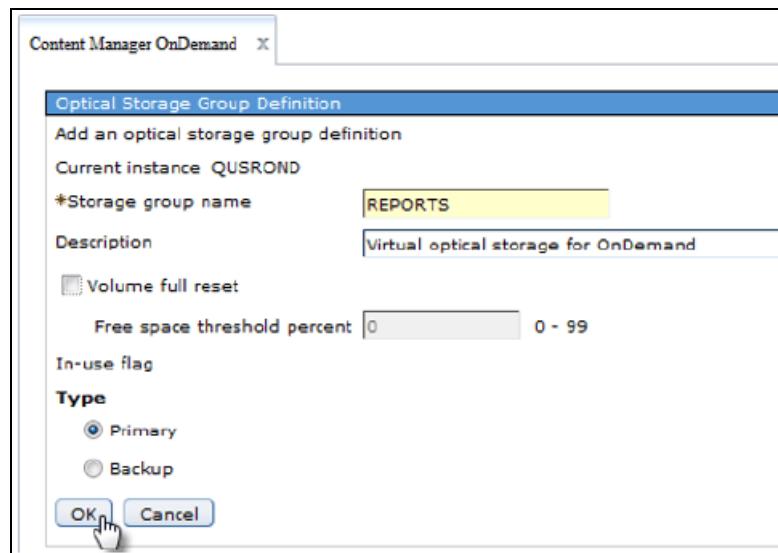


Figure 5-19 Content Manager OnDemand for i optical storage group definition

When you define the optical storage group, you provide the following information:

- ▶ Storage group name
- ▶ Description of the storage group
- ▶ Volume full reset when optical volumes are rewritable and you want to reuse the storage space (only available with local area network (LAN)-attached optical jukeboxes)
- ▶ Free space threshold percent (the percent at which Content Manager OnDemand starts storing to rewritable volumes again if the volume full reset parameter is checked)
- ▶ Storage group type, which is primary or backup

After you define the optical storage group, use IBM Navigator for i to define the optical volumes to the Content Manager OnDemand system (Figure 5-20).

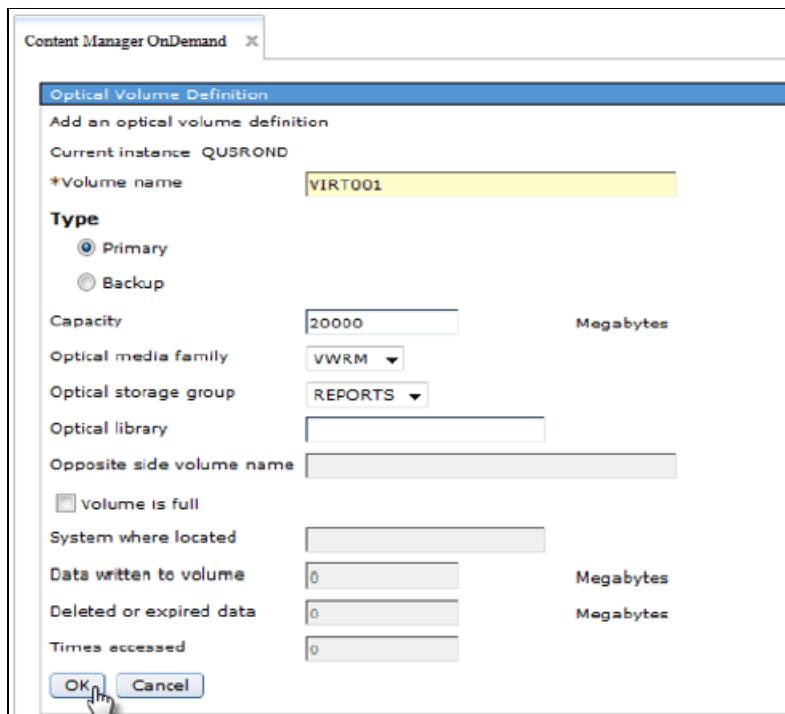


Figure 5-20 Content Manager OnDemand for i optical volume definition

When you define optical volumes, provide this information:

- ▶ Volume name: Your volume name.
- ▶ Volume type: Primary or backup.
- ▶ Capacity in megabytes: Capacity of one side of the optical media after it is initialized.
- ▶ Optical media family:
 - Rewritable (REWT)
 - WORM
 - Universal Disk Format single-sided (UDF1) that is used by DVD RAM drives
 - Universal Disk Format or double-sided (UDF2)
 - Virtual Rewritable (VRWT)
 - Virtual WORM (VWRM)
- ▶ Optical storage group: Your optical storage group.
- ▶ Optical library: Library name, which can be provided for documentation.

- ▶ Volume is full: Set when the optical volume reaches its capacity.
- ▶ Opposite side volume name: For the other side of the optical platter (only required for REWT, WORM, and UDF2).

After the storage groups are established, use IBM Navigator for i to define the migration policy that is needed to use the storage groups (Figure 5-21).

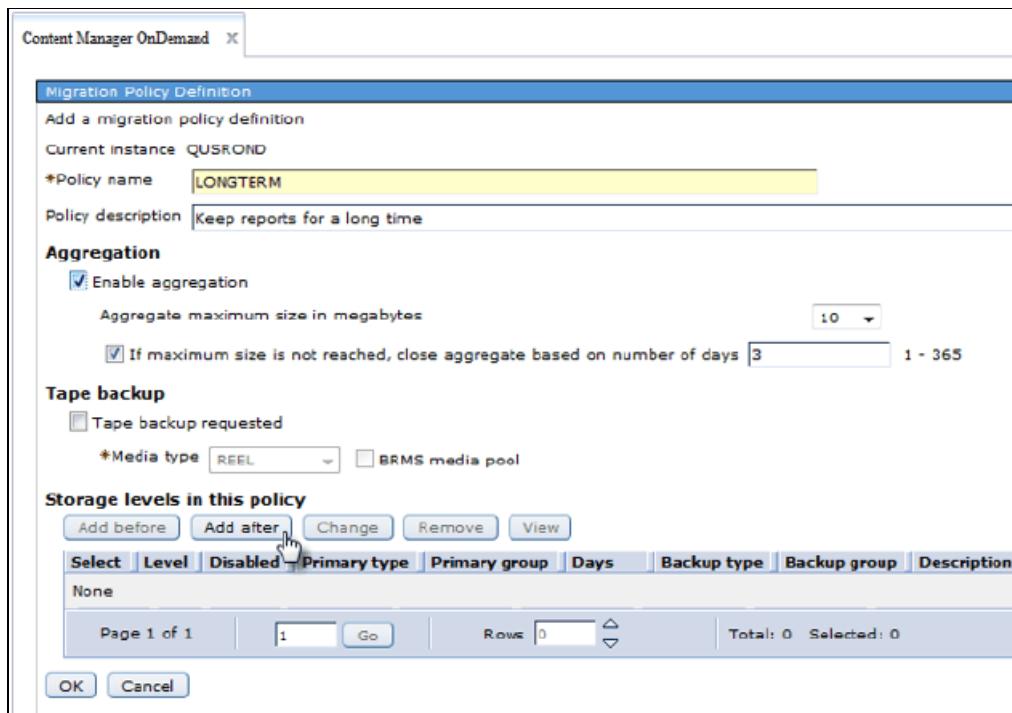


Figure 5-21 Content Manager OnDemand for i migration policy definition

The migration policy definition includes the following items:

- ▶ Policy name and description: This field is for the policy name and its description.

Tip: The preferred practice is to put information, such as the length of time and the location of the data, in the description rather than in the policy name field. You can change, add, and delete levels, but you cannot change the name. If your requirements change, you do not want a name that is no longer accurate.

- ▶ Enable aggregation: If this item is selected, ASM combines individual archived objects into larger objects to provide a more efficient process. Archived objects are appended to the same file until the aggregate is closed.
- ▶ Maximum size: The value of this field determines the maximum size of the aggregate file. ASM closes the existing aggregate and opens a new aggregate when the maximum value is reached.
- ▶ Close aggregate only when maximum size is reached: If this item is selected, the aggregate stays open until the maximum size is reached.
- ▶ Close aggregate based on number of days: The value in this field specifies the number of days before an aggregate closes. ASM closes the aggregate after the specified number of days or when the specified maximum size is reached, whichever occurs first.

Important: The aggregation process occurs before the migration of the object from disk to the first ASM storage level. Only aggregate files that are closed are eligible for migration by ASM. If the specified maximum file size is large and the size of the archived objects is small, the aggregate file can remain open for long periods. Also, the Content Manager OnDemand objects might remain on disk longer than the period that is specified by the application group. Choosing to close the aggregate after a specified period addresses this problem.

- ▶ Tape backup requested and media type: The Tape backup requested field indicates whether a one-time tape backup must be made of the data before it is archived. The Media type field indicates the type of tape to use for the backup.
- ▶ Storage levels in this policy: This section determines the path that the archived data follows through the different archive storage media. The order of the levels determines the migration sequence. Storage levels are created by placing the cursor on an existing storage level (if one exists) and clicking **Add Before** or **Add After**. The Migration Policy Storage Level Definition window (Figure 5-22) opens.

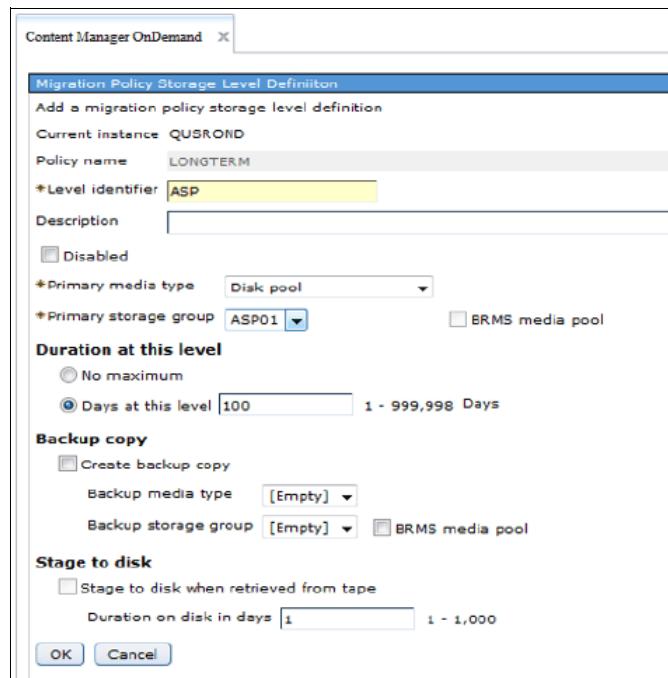


Figure 5-22 Content Manager OnDemand for i new policy level

In the Migration Policy Storage Level Definition window (Figure 5-22 on page 124), you provide the following information for the new policy storage level:

- ▶ Level identifier: This field distinguishes the different storage levels within the migration policy. The value must be unique within the storage levels of the migration policy. ASM uses the level identifier to determine the current level of the migration hierarchy and to determine the next level to which the data must be moved. The identifier can be numeric (for example, 10, 20, and 30) or descriptive (for example, ASP or OPT).
- ▶ Disabled: Specifying this option causes ASM to skip this level in the storage hierarchy. The Disabled option can be used in a situation where an optical unit is added to the system later, but the administrator wants to add an optical policy level and disable it.

This option can also be used when migration to a policy level is discontinued, such as a tape unit. A policy level cannot be removed if data is archived to it, but it can be disabled so that no more data is migrated to that level.

- ▶ Description of the policy level: Use this field to provide a description of the policy level.
- ▶ Primary media type: The types from which you can choose are optical, tape, disk, or expire. If you select expire as the last policy level, when data reaches this level in the migration sequence, it is removed from the archive system even if the retention period that is specified in the application group is not exceeded. It is not necessary to specify an expire level. Instead, you can let the data expire when it exceeds the number of days that are specified in the Life of Data and Indexes in the application group.
- ▶ Duration at this level: In this field, you specify either no maximum or a specified number of days before ASM moves the data to the next level in the migration sequence.
- ▶ Primary storage group: Select the storage group that you want to use to store the data at this level.
- ▶ Create backup copy and backup storage group: You select these options if you want ASM to create a backup copy of the data when it moves to this policy level. The backup storage group must be created with a media type of backup.
- ▶ Stage to disk when data is retrieved from tape and the duration on disk in days: Choose these options to cache data that is returned from tape to disk for the number of days that are specified.

In our scenario, we created a policy level that stores data for 100 days on disk by using the disk pool storage group that is assigned to auxiliary storage pool 1. We also created a policy level that stores data on optical storage indefinitely and uses the REPORTS optical storage group. We did not include an expire level, so the data always expires according to the Life of Data and Indexes in the application group. We can use this migration policy for all application groups if we choose. Documents that are in application groups with the Life of Data set to 100 days or fewer are never migrated to optical because the disk pool storage level specifies 100 days. This approach is easy to manage.

Figure 5-23 on page 126 shows the final migration policy structure.

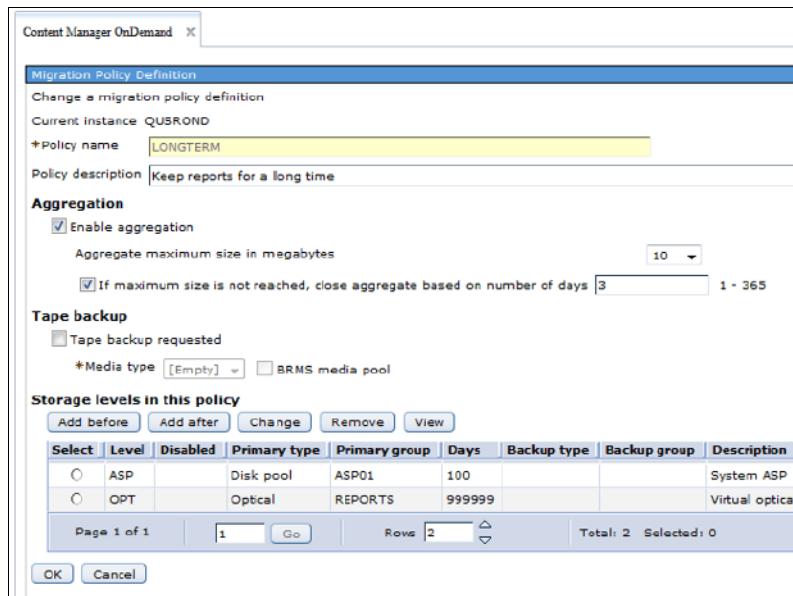


Figure 5-23 Content Manager OnDemand for i migration policy hierarchy

When the migration policy is created, a corresponding storage set is created for the Content Manager OnDemand instance with which the migration policy is associated. The storage set is displayed in a listing of storage sets by using the Content Manager OnDemand Administrator Client but it can only be viewed. No updates can be made to existing storage sets, and no new storage sets can be added by using the Content Manager OnDemand Administrator Client. Storage sets in the Content Manager OnDemand for i system can be created and modified only by using IBM Navigator for i migration policies.

5.4.2 Application group storage management

The application group storage management settings (Figure 5-24 on page 127) determine how long report data and indexes are kept in cache storage before they expire. All documents in the application group are loaded on the media that is part of the storage set to which the application group is assigned. All documents in the application group migrate according to the rules that are defined for the application group's migration policy. When the application group is defined, choices are made concerning how soon data is migrated to archive storage after the report load is completed.

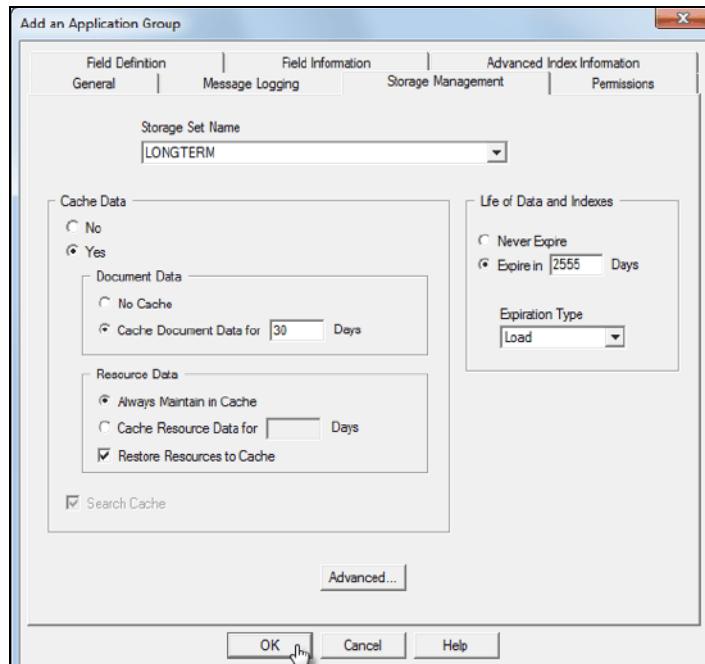


Figure 5-24 Content Manager OnDemand for i Add an Application Group Storage Management tab

Cache Data

The Cache Data setting determines whether the report data is stored in disk cache, and if so, how long it is kept in cache before it expires. If the Cache Data for n Days option is selected, the search cache is always selected.

Search cache determines whether Content Manager OnDemand searches cache storage when users retrieve documents from the application group. When you set Cache Data to No, you can configure Content Manager OnDemand to retrieve existing documents from cache storage while preventing new documents from being copied to cache storage. If you choose not to store reports in cache, you must select a storage set that supports archive storage.

Life of Data and Indexes

The Life of Data and Indexes settings determine the length of time that report data, indexes, and resources are maintained in the Content Manager OnDemand system before they are deleted from the application group. The report data, indexes, and resources can be maintained indefinitely, if set to never expire, or they can be kept for up to 273 years. If your retention requirements change, the Life of Data and Indexes value can be changed. The change affects data that is already archived and new data that is stored to the application group.

Disk Storage Manager maintains documents on disk. It is initiated by the Start Disk Storage Management (**STRDSMOND**) command. Disk Storage Manager can delete documents after they exceed the cache data or Life of Data periods. For more information about running the **STRDSMOND** command, see the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.

Expiration Type

The Expiration Type determines how report data, indexes, and resources are expired. Three expiration types are available:

- ▶ Load: If the expiration type is Load, an input file at a time can be deleted from the application group. The latest date in the input data and the Life of Data and Indexes determines when Content Manager OnDemand deletes the data. Data that is stored in archive storage is deleted by the storage manager based on the archive expiration date. Load is the recommended expiration type.
- ▶ Segment: If the expiration type is Segment, a segment of data, which is a database file that contains index values for an application group, at a time is deleted from the application group. The segment must be closed and the expiration date of every record in the segment must be reached. If small amounts of data are loaded into the application group, and the maximum rows value is high, the segment might be open for a long period, and the data is not expired for the period.
- ▶ Document: If the expiration type is Document, a document at a time is deleted from the application group. Storing with an expiration type of Document causes the expiration process to search through every document in the segment to determine whether the expiration date was reached, resulting in long processing times.

Expiration Type Load is not allowed when you use the ARSLOAD ADD API or when you use the workstation APIs. If you plan to use these APIs with an application group, specify Document for the Expiration Type.

5.4.3 Advanced application group storage management

With the advanced storage management settings (Figure 5-25), you can adjust the size of the load object and determine when report data, indexes, and resources are migrated to archive storage.

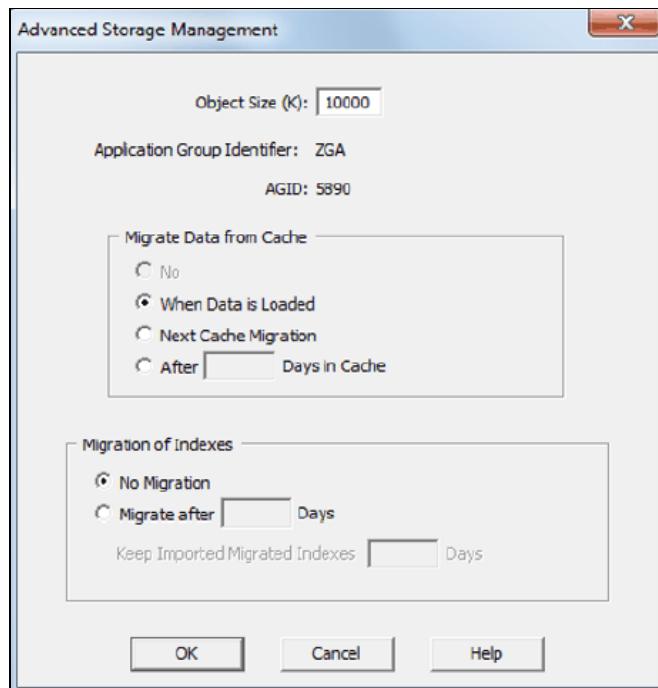


Figure 5-25 Content Manager OnDemand for i Application Group Advanced Storage Management

Object Size

The Object Size parameter determines the size of a storage object in kilobytes. Content Manager OnDemand, by default, segments and compresses stored data into 10 MB storage objects. The default of 10 MB is the recommended object size value.

Important: Setting the value too small or too large can adversely affect load performance.

Note: The object size, which is defined here, must be equal to or larger than the size of the compressed storage objects that are defined in any application that is assigned to the application group.

Migrate Data from Cache pane

This section of the Advanced Storage Management window determines when documents and resources are migrated to archive storage. A storage set that is associated with a migration policy that uses archive media must be selected to enable migration to archive storage. The possible values are listed:

- ▶ No: Data is never migrated from cache. This option is unavailable when a storage set that is associated with archive storage is selected for the application group.
- ▶ When Data is Loaded: Data is migrated to archive storage when the load process runs because of a store command, such as Add Report (**ADDRPTOND**), Start Monitor (**STRMONOND**), or **ARSLOAD**.
- ▶ Next Cache Migration: Data is migrated to archive storage the next time that Disk Storage Manager is run.
- ▶ After Days in Cache: This value specifies the number of days that data remains in cache storage. After the data reaches the prescribed number of days in cache storage, the data is copied to archive storage the next time that Disk Storage Manager is run.

ASM is started with the **STRASMOND** command. The command must be run only in batch. For more information about running the **STRASMOND** command, see the *IBM Content Manager OnDemand for i - Common Server Administration Guide*, SC19-2792.



Security

This chapter describes the security features that are provided by IBM Content Manager OnDemand (Content Manager OnDemand). It also provides examples of available components and their usage to create a secure environment.

In this chapter, we cover the following topics:

- ▶ Content Manager OnDemand security overview
- ▶ Data separation
- ▶ API access
- ▶ Data security
- ▶ Data encryption
- ▶ Security exits

6.1 Content Manager OnDemand security overview

The amount of security that is employed by an organization varies by organization and is normally proportional to the cost of data loss because of security leaks or other issues.

For any system, the first layer of security is its environment. Several attributes are included in a secure environment:

- ▶ Physical security: Controlling physical access to the system and ensuring that the system is protected from both natural and man-made disasters.
- ▶ Data security: Controlling access to online data by using both authentication and authorization techniques; controlling access to offline data, including all backup copies of the data, data storage sites, and encryption of the backup copies of data.
- ▶ Personnel security: Hiring trusted employees, limiting employee access based on employee role, and redundancy.

Although environmental security is beyond the scope of this book, it is important to be aware of and prepare for security in these areas.

This section describes what Content Manager OnDemand can provide from a security perspective.

Content Manager OnDemand is a flexible and scalable system. This flexibility allows the deployment of multiple security features by using multiple methodologies. The descriptions within this chapter are examples of the available components and their usage to create a secure environment.

Figure 6-1 outlines many of the components that are part of Content Manager OnDemand's security features.

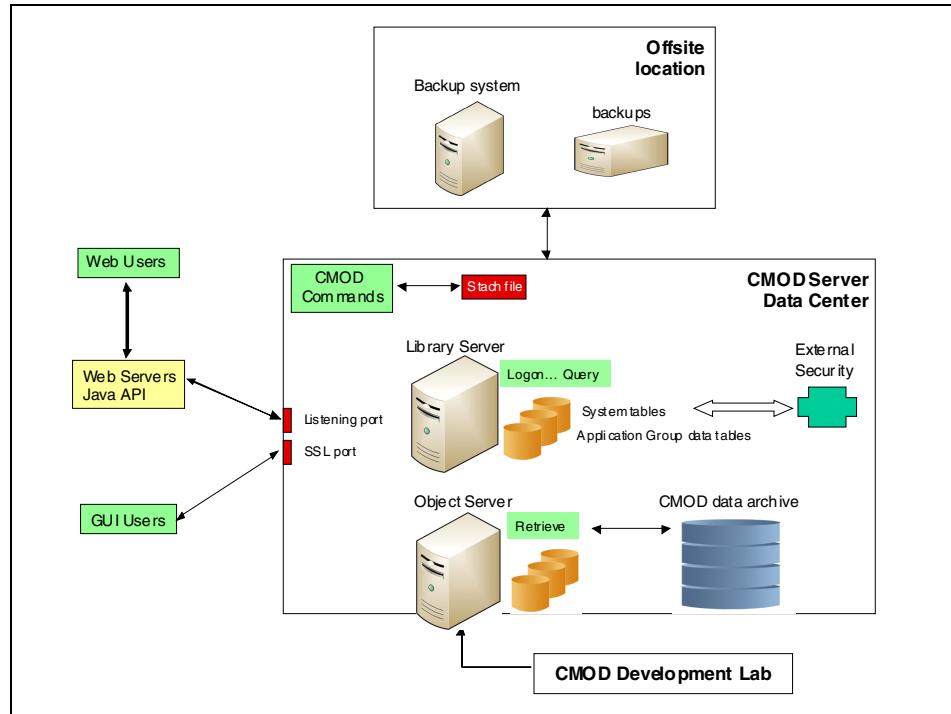


Figure 6-1 Content Manager OnDemand security overview

The complete security cycle begins with code creation through data loading, storage, and access, and ends with data (and index) expiration. The following list outlines different types of security that are described in this chapter. Within each type, different security techniques can be implemented.

- ▶ Code creation:
 - Controlled environment
 - Code scanning
 - Quality assurance testing
- ▶ Data separation:
 - Multiple systems: Allowing users access only to the system that contains data that is relevant to them
 - Multiple object servers
 - Multiple archive subsystems
 - Application programming interface (API) access: Web server, web services, and Content Management Interoperability Services (CMIS)
- ▶ Data security:
 - Administrative features: Login inactivity, disabling a user, locking out a user, and defining password rules
 - Content Manager OnDemand data model: Application groups (AGs) and folders
 - Query restrictions
 - Annotation security
 - Securing access to Content Manager OnDemand commands (stash file)
- ▶ Data encryption:
 - Data at Rest
 - Data in Motion: Secure communication between the server and the clients (Secure Sockets Layer (SSL))
- ▶ Security exits:
 - User security and permissions exit (**ARSUSEC**)
 - Unified logon exit (**ARSPTGN**)
 - System log exit

6.2 Code security

The Content Manager OnDemand code is developed in a secure environment that follows IBM guidelines. The Content Manager OnDemand development lab follows multiple preferred practice methodologies to ensure the highest possible code and security standards. The goal is to ensure that the code “works as designed”, does not perform any malicious actions, and is resistant to external security breach attempts. In this section, we describe examples of the practices that are followed.

6.2.1 Controlled environment

During development, all code is reviewed by two or more developers and passes through a structured process within the development organization to ensure the integrity of the code. The code is periodically scanned to ensure that no foreign code is included and to ensure that safe programming techniques are applied. The following practices are applied:

- ▶ Limited access: The source code is only accessible to the Content Management OnDemand team.
- ▶ Secure systems: The code is stored on secured systems behind the IBM firewall and can only be accessed by the Content Management Development team.
- ▶ Code reviews: All code modifications are reviewed by two or more developers on the team.
- ▶ Separation of duties: Separate development, build, and test teams exist.
- ▶ Redundancy: Two or more developers are familiar with each aspect (function and module) of the code.

6.2.2 Code scanning

The code is scanned three times, once at the beginning of the release or fix pack, once during the middle of the development process, and the last time at the end of the development process. Each time, three types of scans are performed:

- ▶ Code scan: This type of scanning searches for code that is external to the Content Manager OnDemand developed code. The goal is to ensure that no code is unknowingly inserted into the source code and to verify that all of the external code that is used is correctly licensed and will not result in any future legal action.
- ▶ Appscan source: This type of scanning searches for “bad code”. It verifies that all variables and pointers are correctly initialized, and that during the program operation, the values of variables and pointers can be changed only through the “correct” code path and cannot be altered by external sources.
- ▶ Appscan Web: This penetration testing program is run against the common gateway interface (CGI) code to identify any potential security flaws.

6.2.3 Quality assurance testing

The quality assurance (QA) testing is run in parallel with the code development through the development cycle. When developers create new code, they perform their own unit test to ensure that the code works as intended. These unit tests are then passed to the QA team for automation. The QA team automates these tests and adds other newly automated tests to the regression bucket.

Every time a new build occurs, which is nightly during peak development, automated regression and performance tests are run. These automated tests are run on the multiple operating systems that are supported by Content Manager OnDemand (Windows, Linux, AIX, IBM i, Linux on System z®, and z/OS). The goal is to detect any defects or performance impact so that it can be corrected the following day.

Periodically, endurance and stress tests are run to ensure that the code can run for long periods and under heavy workloads.

A specialized subset of these tests and cloud-specific tests are run against the Cloud release of Content Management OnDemand.

6.3 Data separation

Content Manager OnDemand allows the separation (compartmentalization) of the organization's data into multiple separate partitions. Specific groups of users can access only the partitions that contain data that relates to their operations. The separation of data can be at the system level, the object server level, and the archive server level.

6.3.1 Multiple systems

The organization's data can be spread over two or more separate systems. As illustrated in Figure 6-2, User Group A can access only Content Manager OnDemand server A and cannot access any other Content Manager OnDemand system or any other Content Manager OnDemand data. If necessary, you can create a super user group that can access multiple systems.

System access restrictions can be implemented by one or more of the following means:

- ▶ A web server.
- ▶ Firewalls, switches, or other network devices.
- ▶ Only the correct user group is authenticated to use the system.

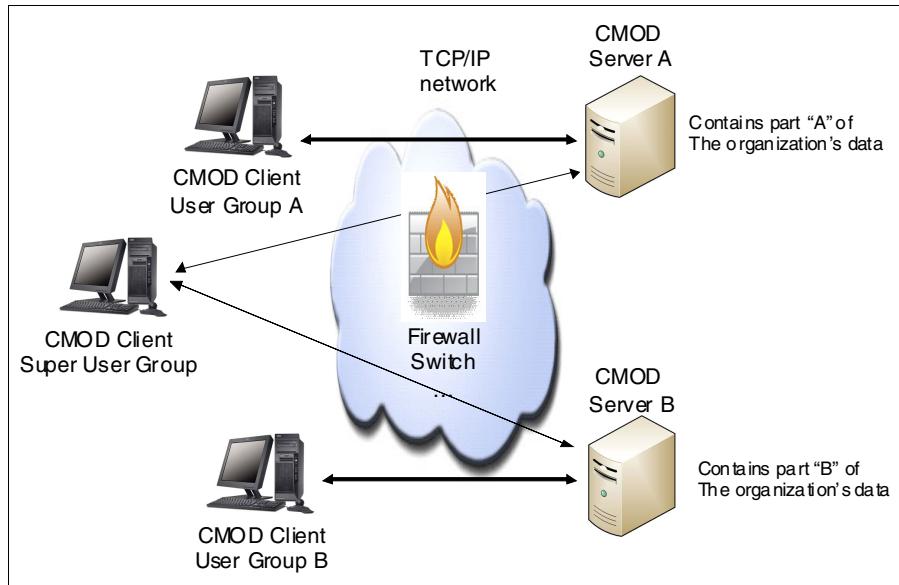


Figure 6-2 Data separation at the system level

6.3.2 Multiple object servers

Data can also be separated at the object server level. In this case, the application group (AG) data tables that contain the indexes that point to separated data are also separated. Therefore, access to the AG data table is allowed only to users who need that data. As illustrated in Figure 6-3 on page 136, User Group A of AG Data Tables Part A is pointed to (allowed access to) the data on Object Server A, and User Group B of AG Data Tables Part B is pointed to the data on Object Server B.

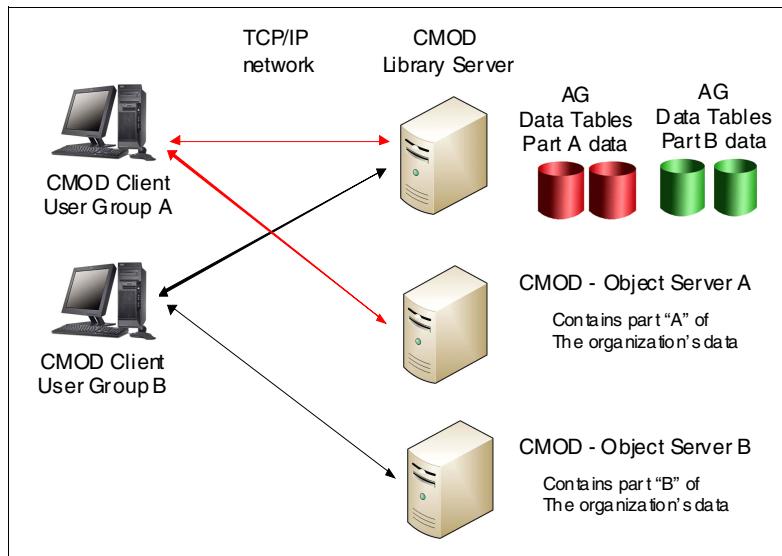


Figure 6-3 Data separation at the object server level

6.3.3 Multiple archive servers

Data can be separated at the archive level. Typically, in this implementation, as illustrated in Figure 6-4, the application group (AG) data tables remain separate and User Group A's data is stored on the Tivoli Storage Manager system A server, and User Group B's data is stored on the Tivoli Storage Manager system B server. The two Tivoli Storage Manager servers are separate systems. This same type of separation is also possible by using object access method (OAM) on z/OS systems. OAM enables the separation of data by placing the data in different OAM collections on different storage devices.

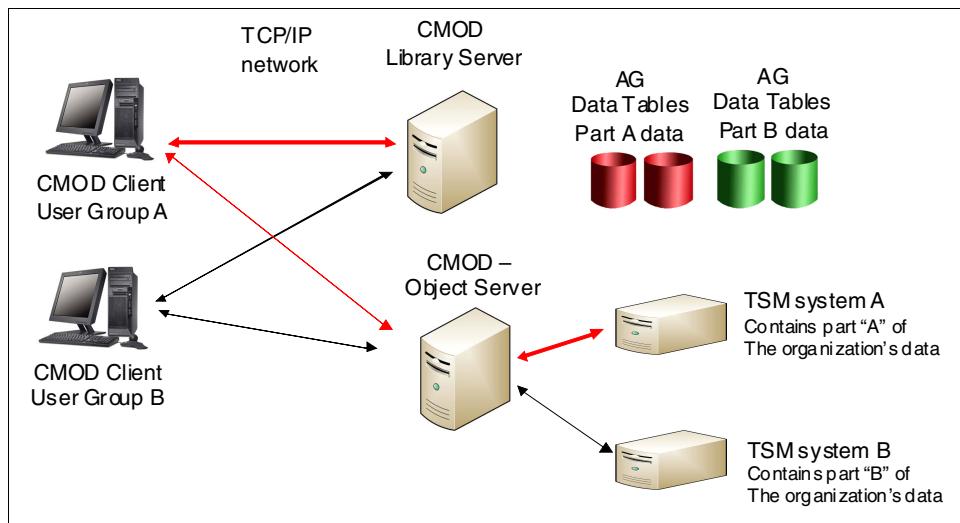


Figure 6-4 Data separation at the archive level

6.4 API access

An important component of Content Manager OnDemand is the Content Manager OnDemand Web Enablement Kit (ODWEK) Java APIs. These APIs are used to build applications that access the Content Manager OnDemand server. Various applications can be built by using the APIs. Examples of applications include IBM Content Navigator (ICN) and CMIS. By using the ODWEK APIs, you can also build your own application server or web services applications.

All of these types of applications address the following situations:

- ▶ Users communicating and interacting with a mid-tier, custom-built access mechanism that controls access to the Content Manager OnDemand server. For example, the mid-tier application can control whether a Content Manager OnDemand user request is accepted or rejected, and if it is accepted, which Content Manager OnDemand server the request is routed to.
- ▶ The network transmissions between the ODWEK Java APIs and the Content Manager OnDemand server use a proprietary Content Manager OnDemand protocol and optionally can be encrypted by using SSL.
- ▶ The network transmissions between the mid-tier custom application and the users can optionally be encrypted by using SSL.
- ▶ By using an optional user proxy implementation, multiple users can share a user ID and password, therefore reducing the number of actual logons to the Content Manager OnDemand server while maintaining secure access to the system through the custom-built access mechanism.
- ▶ The Java APIs can pass a security token through to the Content Manager OnDemand server. This token can then be captured by the security exit and the exit can perform the required special processing.

Figure 6-5 shows controlling access at the web server.

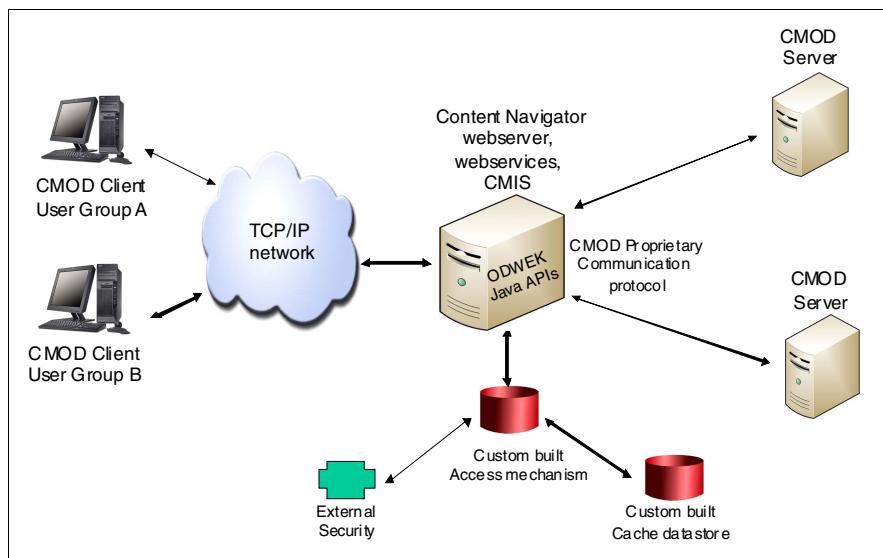


Figure 6-5 Data separation at the web server (mid-tier)

6.5 Data security

Access to the Content Manager OnDemand data tables is secured through various methods. These methods include a secure data model, user authentication, SQL Query support, annotation security, and securing access to the Content Manager OnDemand commands. These methods are described in further detail in this section.

6.5.1 Content Manager OnDemand object-owner model

Content Manager OnDemand internal security is based on an object-owner model, which is illustrated in Figure 6-6. Details about the object-owner model are in the *IBM Content Manager OnDemand for Multiplatforms, V9.5, Administration Guide*, SC19-3352. In this context, a Content Manager OnDemand instance is an implementation of the library server, one or more object servers, the data access, and the storage model. The data access and storage are implemented in the form of objects. The following objects are all Content Manager OnDemand objects:

- ▶ Users
- ▶ Groups
- ▶ Application groups
- ▶ Folders
- ▶ Cabinets
- ▶ Applications
- ▶ Holds
- ▶ Storage set
- ▶ Printers

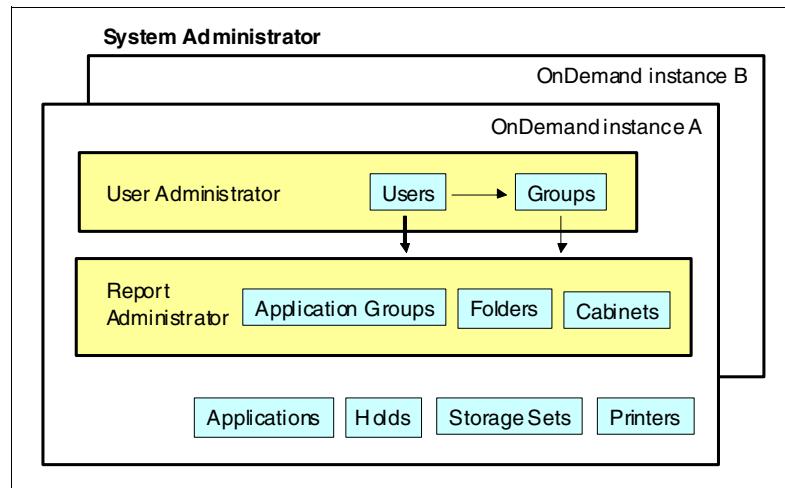


Figure 6-6 Content Manager OnDemand internal security

The Content Manager OnDemand object-owner model design handles the following situations:

- ▶ A single system administrator to control one or more Content Manager OnDemand instances through a single Administrator Client interface.
- ▶ Flexibility to create user administrators who manage users and groups for a specific Content Manager OnDemand instance.
- ▶ Flexibility to create report administrators who manage application groups, folders, and cabinets for a specific instance.

- ▶ Implementing report security that is based on limiting object access to selected groups of users.
- ▶ Elimination access to Content Manager OnDemand objects unless explicit permission is granted.

In summary, with this model, organizations can separate and isolate report (data) ingestion and access to various users and groups. Additionally, organizations that provide billing, payroll, accounting, and bill presentment services for a number of other companies (their clients) also benefit from this model, because users from one company are isolated from the data and users of another company. Furthermore, large systems can decentralize system administration so that report and user administrators can be delegated for the management of components of the overall Content Manager OnDemand system.

6.5.2 Administrative features

Use the Administrator Client to control user logon parameters. These parameters are set in the Login Information tab in the System Parameters window, as shown in Figure 6-7.

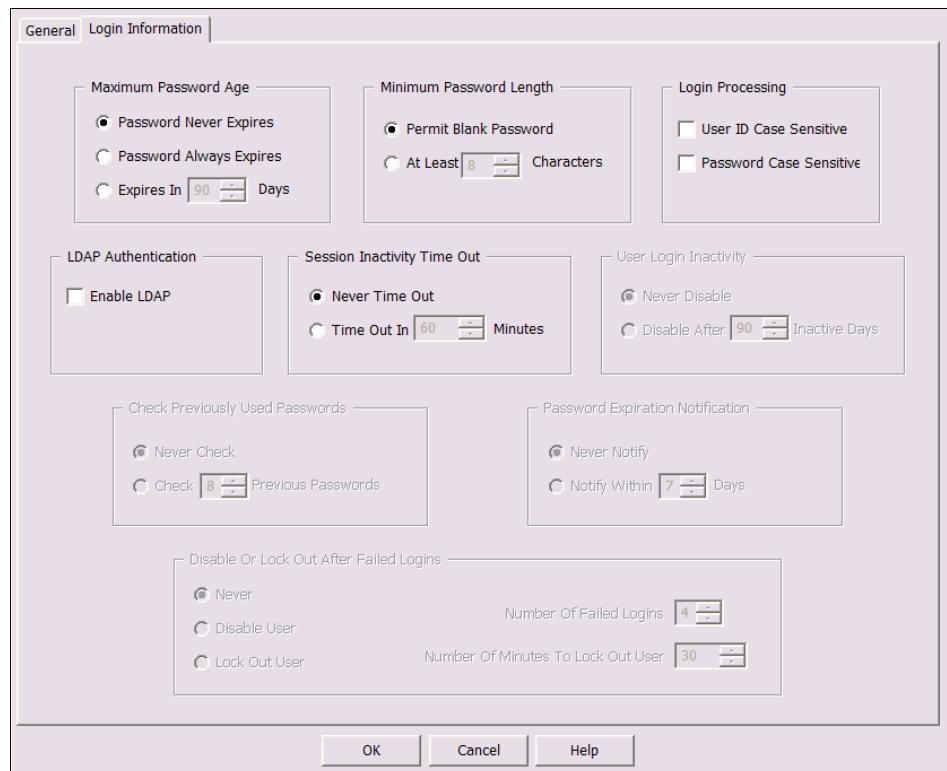


Figure 6-7 Administrator Client - setting the logon restrictions

We describe these parameter settings in the following subsections.

Check Previously Used Passwords

This setting specifies whether you want users to be able to reuse a previous password. You can make users create up to 10 unique passwords before they can reuse a previous password. Use this setting to enforce security policies. For example, you can force the user to not reuse the eight most recent passwords.

Disable Or Lock Out After Failed Logins

This setting specifies whether you want to limit the number of failed login attempts by a user. You can limit the number of login attempts, specify how many failed attempts you want to permit, and specify whether to disable or lock out the user after the user exceeds that number of attempts.

If you choose to disable a user, the user must request that the system administrator re-enable the user ID.

If you choose to lock out the user, the user must wait to attempt another login. You specify how many minutes to wait in the Number Of Minutes To Lock Out User field.

LDAP authentication

Use Lightweight Directory Access Protocol (LDAP) to store authentication values on a separate organizationally centralized server that is remote from Content Manager OnDemand. LDAP can be used in place of the user security exit to manage basic login authentication. Figure 6-8 shows how Content Manager OnDemand works with LDAP.

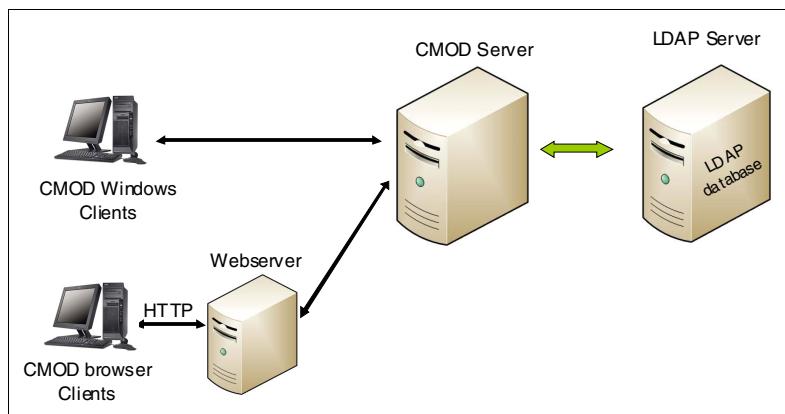


Figure 6-8 Content Manager OnDemand works with LDAP

You can specify whether you want to use LDAP authentication in your Content Manager OnDemand server.

When you enable LDAP authentication, the Content Manager OnDemand server makes an authentication request to the LDAP server every time it receives a login request from the client. The Content Manager OnDemand server processes the client request only after the user information is verified by the LDAP server.

If you use LDAP, consider the following scenarios:

- ▶ The LDAP server runs on another system and it connects to Content Manager OnDemand through TCP/IP.

In this scenario, a time delay occurs between when the verification request is issued by Content Manager OnDemand and the result is returned to Content Manager OnDemand. The length of this time depends on the Internet Protocol network connection, the response time of the LDAP server, and the current LDAP workload.

- ▶ Users with admin-level security bypass LDAP.

You can compare an admin user's response time to a production user's response time to determine the LDAP impact.

- ▶ The LDAP server or the connection to the LDAP server fails.

When this scenario happens, users cannot log in to Content Manager OnDemand, except for users with admin-level security.

Login Processing (case sensitivity)

With this parameter, you can specify whether user IDs and passwords are case-sensitive. By default, user IDs and passwords are *case-insensitive*. When you add a user, Content Manager OnDemand converts lowercase letters in the user ID to uppercase letters.

A person can type letters in a user ID in uppercase, lowercase, or mixed case letters. For example, if you add the user LaGuarde, a person can enter LAGUARDE, lagarde, or LaGuarde to log on to the server.

If you select User ID to be case-sensitive, a user must type the user ID exactly as it was entered when the user was added. For example, if you add the user ID LaGuarde, the user must enter LaGuarde to log on to the server.

If you set a password as case-sensitive, a user must enter the password exactly as it was entered when the user was added.

Important: *Do not change* the case-sensitive settings for user IDs and passwords after you install the system.

Decide whether to make user IDs and passwords case-sensitive *when* you install the system. Change the defaults if necessary, but do not change the settings later. Otherwise, the following situations occur:

- ▶ If user IDs are initially case-insensitive and you later choose User ID to be case-sensitive, user IDs that were added before you changed the parameter must be entered in uppercase. The same is true for passwords.
- ▶ If user IDs are initially case-sensitive and you later clear the case-sensitive restriction, the user IDs that were added before you changed the parameter might contain mixed or lowercase letters, which are no longer valid. The same is true for passwords.

Note: If users log on to Content Manager OnDemand with the IBM CICS® client program, you must configure the system to ignore the case of user IDs and passwords.

Maximum Password Age

This setting specifies a time limit for passwords and determines when Content Manager OnDemand prompts users to change passwords. The default setting is Password Never Expires, which means that passwords do not expire and Content Manager OnDemand never prompts users to change passwords.

If you click **Password Always Expires**, users must change to new passwords each time that they log on to a server. To set a specific time limit for passwords, select **Expires In __ Days** and enter the number of days that passwords are valid in the space that is provided. The value can be 1 - 365.

Minimum Password Length

This setting specifies whether passwords are required. If passwords are required, it specifies the minimum number of characters that passwords can contain. The default value is At Least 8 Characters, which means that passwords must contain at least eight characters.

If you click **Permit Blank Password**, which means that passwords are not required, the valid password length is 0 - 128.

To set a specific minimum password length, click **At Least __ Characters** and enter a number in the space that is provided. The value can be 1 - 128.

When a user changes a password, the client checks the number of characters that the user entered. The new password must contain the minimum number of characters. Otherwise, the user receives an error message.

Password Expiration Notification

This setting specifies whether to notify users that their password expires within the specified number of days.

Changing an Expired Password

Content Manager OnDemand provides password expiration processing to help you manage security on the system. You can set a value that represents the time in days that passwords that are assigned to users remain valid. After a user's password reaches the value that you specify, the user must change the password.

After a password reaches the expiration value, the next time the user logs on to a server, Content Manager OnDemand prompts the user to enter a new password. The user must enter the current password, a new password, and verify the new password by reentering the new password.

Session Inactivity Time Out

This setting specifies when Content Manager OnDemand terminates sessions between inactive clients and the server. The default setting is Time Out in 60 Minutes. Never Time Out means that Content Manager OnDemand does not terminate a session, regardless of how long the client remains inactive.

To set a specific inactivity timeout, click **Time Out In __ Minutes** and enter the number of minutes in the space provided. The value can be 1 - 1440 (24 hours). The period of inactivity is measured between requests to a server. For example, when a user enters a query, Content Manager OnDemand searches the database and builds the document list. This action completes a request to the server. If the user does not work with the items in the document list, open another folder, or start another query before the inactivity timeout occurs, Content Manager OnDemand automatically terminates the session with the client.

Use caution when you set the inactivity timeout. Choose the correct amount of time when you specify this setting. For example, assume that you set the inactivity timeout to 10. You log on to Content Manager OnDemand to add an application group. Creating the application group might take you 15 minutes to complete. After you enter all of the information about the application group, you click **OK** to create the application group. Content Manager OnDemand issues a message that a timeout occurred. You must log off the server, and you cannot save the information that you entered about the application group.

System Logging

This setting specifies the messages that Content Manager OnDemand saves in the system log. Content Manager OnDemand provides the system log to help you track activity and monitor the system. Content Manager OnDemand saves messages that are generated by the various programs, such as the **ARSLOAD** program. Content Manager OnDemand can save a message in the system log when the following events occur:

- ▶ A user logs on to the system.
- ▶ A user logs off the system.
- ▶ A user logon fails.
- ▶ Application group data is queried, retrieved, loaded, updated, deleted, or maintained.

System Log Comments

This setting specifies whether the Administrator Client displays the System Log Comments window when you perform an add, update, or delete operation.

You can enable comments and also specify whether the comments are required. If the comments are required, the user must enter one or more characters in the Comments field.

User Login Inactivity

This setting specifies whether you want to disable users who do not log in after the specified number of days. Users must contact the system administrator to enable their user IDs.

Query Restriction

This setting specifies the restriction to access to folders and application groups based on index values. This setting is specified on the Permissions tab of the Update an Application Group window, as shown in Figure 6-9 on page 144. You can set a restriction with the internal Content Manager OnDemand security. The access restriction for an application group is controlled through internal or external permissions (for example, RACF).

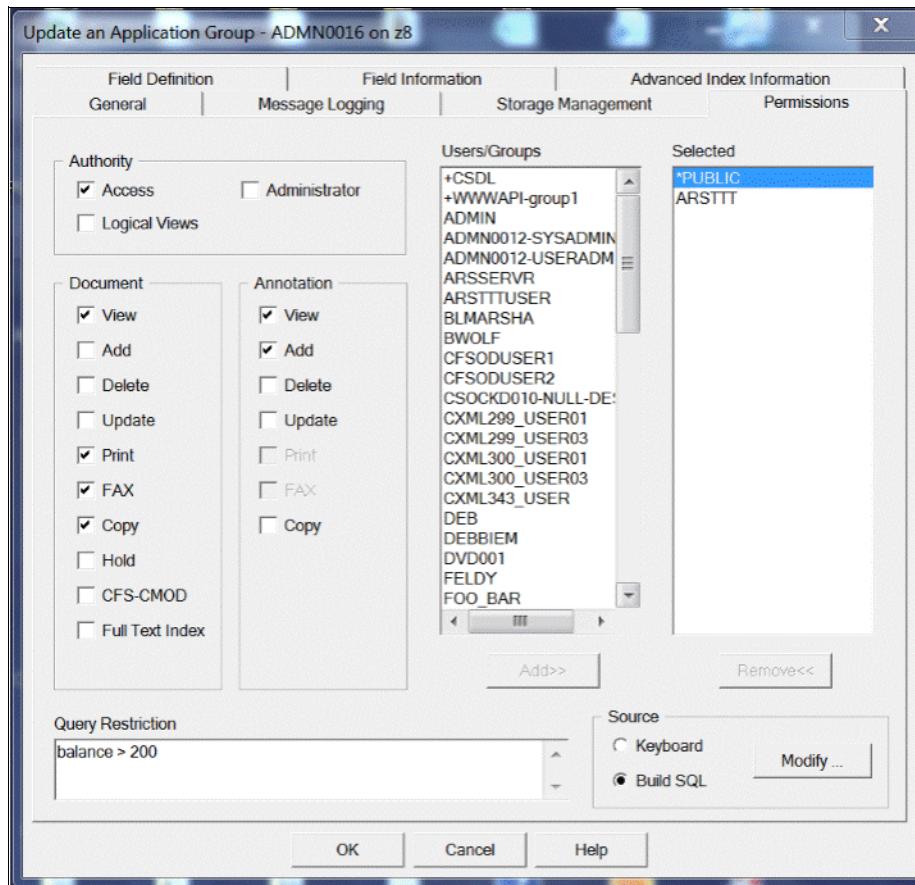


Figure 6-9 Update query restriction

When a user is given access to the application group, access can be further restricted to a subset of the application group data by using a query restriction setting on the application group. The query restriction is added to an SQL “where clause” that enforces the restriction.

Figure 6-9 is an example of a query that is restricted to statements with a balance that exceeds 200. This query restriction is for all users with access to the application group (*PUBLIC) that do not have a separate query restriction.

6.5.3 SQL macro support

Macro support can be used in SQL statements, including the query restriction. With the macro support, the macro can be substituted by the appropriate value for the creation of SQL statements that include current object values, such as application group name or user ID. The available macros are listed in Table 6-1 on page 145.

Table 6-1 Available macros

Name	Description
\$ODUSERID	The user ID that is used to log in to Content Manager OnDemand.
\$ODALIAS	The alias that is defined to Content Manager OnDemand for the user's session.
\$ODAGNAME	The application group name.
\$ODAGID	The application group internal identifier.

The substitution does not include any necessary quotes for the macro, so you must ensure that you use the correct quotation marks for the macro, if required, for example:

```
WHERE ag_field in (SELECT value FROM <customer_table> where userid = '$ODUSERID')
```

If you log on to Content Manager OnDemand as USER1, the SQL changes to the following syntax:

```
WHERE ag_field in (SELECT value FROM <customer_table> where userid = 'USER1')
```

6.5.4 Annotations security

Content Manager OnDemand allows the secure creation and viewing of annotations (notes). This capability is enabled through the Administrator Client window, as shown in Figure 6-10.

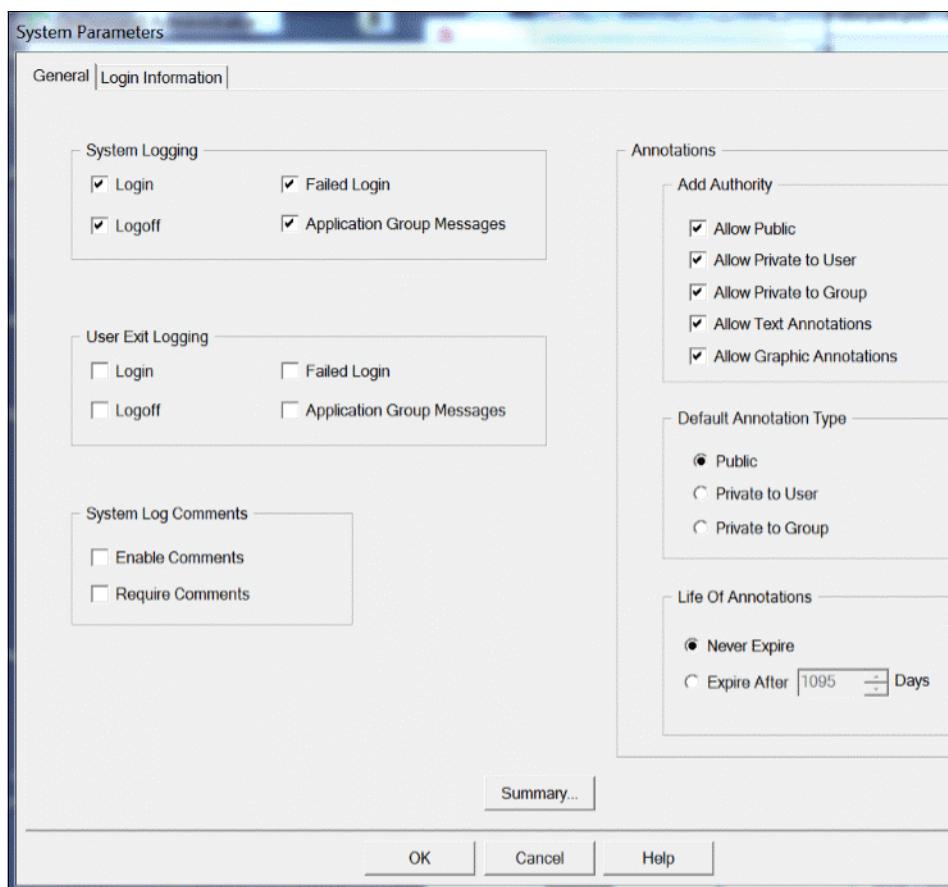


Figure 6-10 Add annotation authority

Controlling annotation creation

In Figure 6-10 on page 145, in the Add Authority section, specify the types of annotations (referred to as “notes” in Content Manager OnDemand Client) that can be added by a user. This selection applies to all users with authority to add annotations in the system.

You can select the following types of annotations:

- ▶ Allow Public: Allows the user to add public annotations. Public annotations of a document can be viewed by anyone who opens that document.
- ▶ Allow Private to User: Allows the user to add private annotations to a document. These annotations can be viewed only by the user that created the note, application group administrators, and system administrators.
- ▶ Allow Private to Group: Allows the user to add annotations to a document. These annotations can be viewed only by a specific group of users.
- ▶ Allow Text Annotations: Allows the user to add text annotations.
- ▶ Allow Graphic Annotations: Allows the user to add graphic annotations.

Controlling annotations viewing

In the Annotation section of the Permissions tab of the Add an Application Group window, specify the default viewing scope for all annotations, as shown in Figure 6-11.

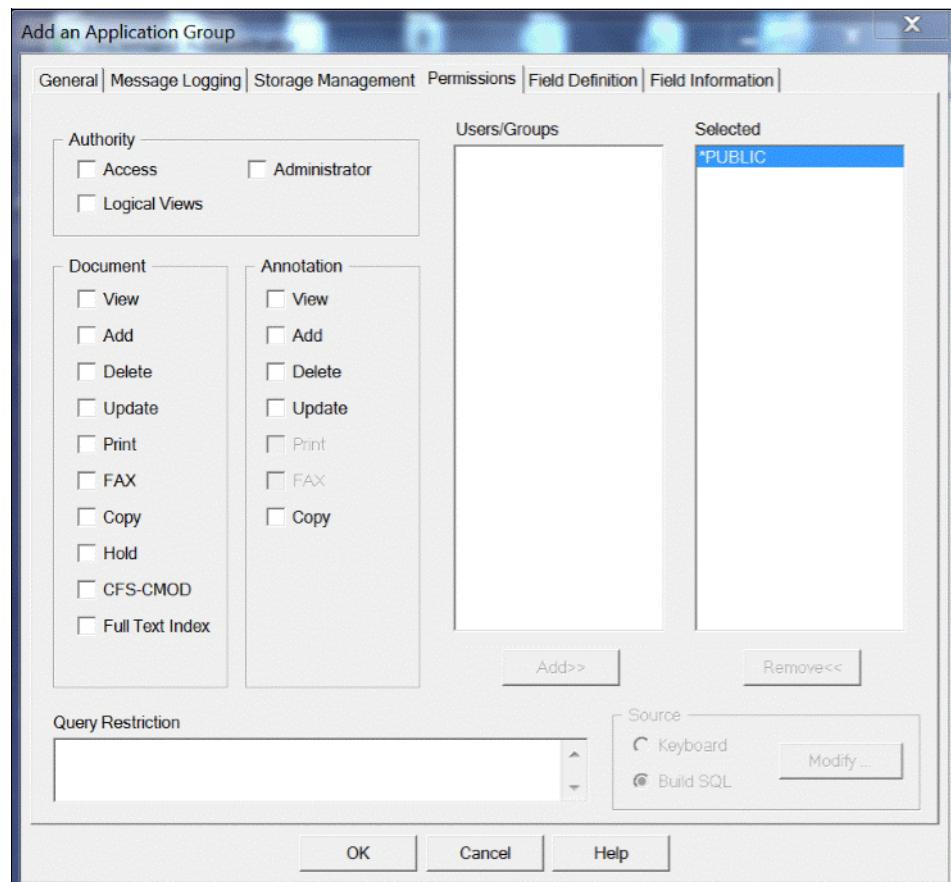


Figure 6-11 Annotation viewing

You can select the following scopes:

- ▶ View: Lets the user view annotations.
- ▶ Add: Lets the user add annotations to documents.
- ▶ Delete: Lets the user delete annotations.
- ▶ Update: For text annotations, lets the user update the location of an annotation on the page (by dragging the annotation marker to a new spot on the page). For graphical annotations, this scope lets the user update the various characteristics of an annotation.
- ▶ Copy: For text annotations, lets the user copy the text of an annotation to the clipboard.

6.5.5 Securing access with ARSSTASH and the stash file

Use the stash files and the **ARSSTASH** command to securely store and pass a password to Content Manager OnDemand commands without the passwords appearing in the clear (unencrypted text). The **ARSSTASH** command is used to encrypt the password by using Advanced Encryption Standard (AES)-128 encryption and storing it in a file that is called a *stash* file (an encrypted password file). The path to that stash file is then specified with the **-p** parameter to those commands that require a password. The stash file is retrieved and decrypted, and the password that is stored in the stash file is used. Therefore, the **-p** parameter that is stored in JCL or other scripts or programs does not need to contain a clear text password.

Multiplatforms: Stash files are the method of choice for securely storing passwords on a Content Manager OnDemand for Multiplatforms server. Unified login does not work when you use a Content Manager OnDemand for Multiplatforms server. Therefore, stash files are the only mechanism that is provided to prevent passwords from being specified in "clear text" for the various Content Manager OnDemand commands that require passwords.

Special case for z/OS and IBM i

If you are using a z/OS server, consider using the z/OS unified login mechanism instead of the stash file. Unified login provides the same functionality as stash files, which means that the passwords are not stored unencrypted when at rest (for example, in JCL or scripts) but without the additional burden of managing stash files. For example, when a password is changed for a user, stash files that contain the encrypted password for that user must also be changed.

If you are using an IBM i server, you might not need to use stash files because if you are signed on to the IBM i server with a user profile that is defined to Content Manager OnDemand and that has enough authority to perform the function you are running, Content Manager OnDemand uses the IBM i user profile for that function (such as **ARSDOC** or **ARSLOAD**). The **-u** and **-p** parameters are not required, therefore relieving you of the need to show or store a password in clear text.

Accessing the stash file

Access to the stash file must be restricted by using file system permissions and other security as appropriate. The stash file that is used by an instance is specified in the **ARS.INI** file (or in the registry on Windows) with the **SRVR_OD_STASH** parameter, for example:

```
SRVR_OD_STASH=/opt/IBM/ondemand/V9.5/config/ars.stash
```

At IBM i version 7.2 and later, the following commands support the optional password stash file (STASHFILE) parameter:

- * ADDRPTOND
- * MRGSPLFOND
- * STRMONOND

Using the stash file

The stash file can be used by these commands:

- ▶ **arsadmin**
- ▶ **arsdoc**
- ▶ **arsload**
- ▶ **arsmaint**
- ▶ **arsrd**
- ▶ **arsxml**

In our example, we use **arsload**. The supported values for the **-a** parameter are available in the **arsstash help** output.

The preferred method is to set a user ID and password for each command in the stash file. Then, the **arsload** command can be run without specifying the **-u userid** or the **-p password** parameter. This method is always recommended when you run the **arsload** command as a daemon. To use this method, first run the **arsstash** command to store the user ID and password for the **arsload** command:

```
arsstash -a 3 -s ars.stash -u <userid>
```

Then, enter and verify the password when you are prompted. When you run **arsload**, omit the **-u** and **-p** parameters. The **arsload** command obtains the **arsload** user ID and password from the stash file.

A second method is to specify the **-u** parameter for another Content Manager OnDemand user ID that exists in the stash file. To use this method, first run the **arsstash** command to store the user ID and password in the stash file:

```
arsstash -a 1 -s ars.stash -u <userid>
```

Then, enter and verify the password when you are prompted. When you run **arsload**, specify the **-u <userid>** and **-p <stash file>** parameters. The **arsload** command obtains the password for the specified user ID from the stash file.

Notes:

- ▶ You can continue to run the **arsload** command with the password in clear text. However, the **arsload** command generates a warning that specifying the password in clear text is being deprecated and to use the stash file instead.
- ▶ The stash file works with Content Manager OnDemand security, LDAP, and IBM i security.
- ▶ After you save the user ID and password in the stash file, remember to change the password anytime that you change the user's password in Content Manager OnDemand; otherwise, the load fails. The **ARSLOAD** program can accept an expired password. However, the **ARSLOAD** program fails if an incorrect password is specified.

Stash file information for z/OS

To use `arsstash` with Content Manager OnDemand for z/OS, the Integrated Cryptographic Service Facility (ICSF) must be available on the z/OS system to provide AES-128 encryption. The encryption can be performed in either software or hardware.

In the examples, a started task name of CSF is used. If CSF is not running, when you try to create a stash file, you get the following message, which does not identify the problem:

Verify OnDemand Password:

```
ARS1602E The stash file >/u/myuser/prodstash.stash< is invalid.  
/usr/lpp/ars/V9R5M0/bin: >
```

To verify that CSF is up and running so that Content Manager OnDemand V9.5 can use it, use the **MODIFY** command against **ARSSOCKD**.

On a system where ICSF is up and running, run the following command:

```
F ARSSOCKD,D,ICSF  
ARS0438I 15.21.18 DISPLAY ICSF  
CSFIQF RC=00, RSN=00000000, AES=3, FMID=HCR7780
```

On a system where CSF is not running, run the following command:

```
F ARSSOCKD,D,ICSF  
ARS0438I 15.28.36 DISPLAY ICSF  
CSFIQF RC=12, RSN=00000000, AES=0, FMID=N/A
```

6.6 Data encryption

Encrypting data is a way of providing security and protection to your Content Manager OnDemand data.

6.6.1 Encrypting data at rest

Depending on how the database tables and archived data are stored, you can encrypt the data by using either DB2 encryption or device encryption. The advantage of encrypting the data is to make it “unintelligible” to unauthorized access even if it is accessed (as an extreme example, the storage device is stolen). The cost of encrypting the data is increased processor consumption and slower response time. This cost varies based on the device and encryption methods that are used.

Backup data must always be encrypted because it is more susceptible to unauthorized access.

6.6.2 Encrypting data in motion: Secure communications

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) allow secure communication between the Content Manager OnDemand server and the Content Manager OnDemand clients. Since Content Manager OnDemand version 8.5, support for SSL and its successor, TLS, is enabled for all transmissions between the Content Manager OnDemand servers and clients. When this section mentions SSL, the same information applies to TLS, unless otherwise noted.

SSL is the standard technology for creating secure connections between servers and clients. The secure connection allows authentication and verification, and data encryption.

Authentication and verification allow both the client and server to verify that they are communicating with the intended receiver. Data encryption prevents the packets of information that are traveling through the network to be viewed by anyone who can access the network.

During an SSL handshake, a client and server securely exchange digital signatures and encryption keys by using a public-key algorithm (usually Rivest-Shamir-Adleman algorithm (RSA)). The client and server establish a secure connection with this identity and key information. After the client and server establish a secure session, they transmit the data to each other, encrypting it with a symmetric algorithm, such as AES.

Trusted parties, which are called certificate authorities (CAs), issue digital certificates to verify the identity of an entity, such as a client or a server. The digital certificate serves the following purposes:

- ▶ Verifies the identity of the owner
- ▶ Makes the public key of the owner available

The IBM Global Security Kit (GSKit) provides libraries for data encryption and SSL communication.

The GSKit package also installs the iKeyman key management utility (gsk7ikm), which you can use to create key databases, public-private key pairs, and certificate requests. For information about the iKeyman utility and the GSKCmd command-line interface, see the *IBM Developer Kit and Runtime Environment, iKeyman 8.0 User's Guide* at the following website:

<http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/security/60/iKeyman.8.User.Guide.pdf>

Note: Implementation of SSL is optional. The Content Manager OnDemand server can be configured to listen on either a non-SSL port or an SSL port, or it can listen on both types of ports. To implement SSL, click **New server**. In the Add a Server window that opens, select **Use Secure Sockets Layer**. If your server does not support SSL, SSL is not used even if you select this check box.

After a Content Manager OnDemand client (for example, the Content Manager OnDemand Windows client, ARSDOC, or OnDemand Web Enablement Kit (ODWEK) Java API) is configured to log on to a Content Manager OnDemand library server with SSL, all communication to and from that client is performed by using SSL:

- ▶ Between the client and the library server
- ▶ Between the client and the object servers

To use SSL, it must be enabled on both the server and the client components of Content Manager OnDemand.

Important considerations exist when you use SSL. We describe them in the following subsections.

Separate port number

In addition to the standard (non-SSL) port, a separate port number is identified on the Content Manager OnDemand server to support the secure connection. This separate port number allows both SSL and non-SSL connections to operate concurrently. When a client connects to the SSL port, it negotiates a connection through a handshake procedure during which the client and server agree on the session parameters to use to maintain a secure connection. Session keys are generated that allow the encryption and decryption of the data that is sent between the client and server.

Processor consumption

SSL improves security by encrypting and decrypting data across the network. The encryption and decryption occur at the application layer, which consumes the additional processing cycles for both the sending and receiving systems. Therefore, consider using SSL only for sessions where it is needed. Consider adding additional processor resources on the Content Manager OnDemand server or clients to manage the increased overhead processing.

Digital certificates

With SSL, the identities of the parties are verified by using digital certificates. Digital certificates have expiration dates. After a digital certificate expires, Content Manager OnDemand will not be able to establish connections through SSL. Therefore, always be aware and plan ahead to avoid expired certificates.

ODWEK

The support of SSL and ODWEK refers specifically to the transfer of data between ODWEK and the Content Manager OnDemand servers and it does not imply a level of support from the browser to ODWEK. The use of SSL from the browser to ODWEK was always allowed and it does not require any support from ODWEK. It is the application and the web developer's responsibility to enable such support.

arsload

GSKIT is initialized one time for each **arsload** invocation. When you load multiple documents, it is more effective to concatenate the documents (such as TIFF images) and generic index files and load multiple documents at a time. Also, when you load multiple documents, use **arsload** as the daemon.

6.7 Security exits

The Content Manager OnDemand security exits allow customers to implement their own customized security methods based on their internal requirements and needs. You can use the security exit to customize and enhance the security functions within a Content Manager OnDemand system.

6.7.1 User security and permissions exits

Content Manager OnDemand provides a user exit so that you can implement your own user exit program to identify and authenticate users that log on to the system. If you use only Content Manager OnDemand internal security, the security exit is not needed.

You can use the security user exit to authenticate a user's password. For example, you might want to enforce a sort of password uniqueness or allow logons to occur only at specific times in the day. You can also build a user proxy mechanism to allow users that are not already in the Content Manager OnDemand user database to access the system.

The permissions exit is called during login if the permissions exit is turned on for folder and application groups. It is also called during a search when the permissions exit is turned on for an SQL query string or document.

Use the user security exit and the permissions exit to augment the security-related processing of the following activities or events:

- ▶ User authentication (checking user security):
 - Log on.
 - Change a password.
 - Add a user ID.
 - Delete a user ID.
- ▶ Resource authorization (checking user permissions):
 - Access to a Content Manager OnDemand folder.
 - Access to a Content Manager OnDemand application group.
 - Restrict access to specific documents.
 - Control the SQL search criteria that are used for searching folders.

The user-written exit routine (or set of exit routines) can interact with another security system to determine whether the activity is allowed or disallowed.

Important: When you implement your own security user exit program, you bypass the logon verification processing that is built into the base Content Manager OnDemand product. We advise caution when you bypass the Content Manager OnDemand user and password restrictions. The security of the system can easily be subverted by malicious or defective code. Only use code that you trust.

When you set the user security exit, set the following parameters:

- ▶ Set the Maximum Password Age parameter to the value that best matches the main logic of the user exit program (permit/deny). The Maximum Password Age parameter is set on the System Parameters dialog box, which is accessed by using the Administrator Client.
- ▶ Set the Maximum Password Age parameter to Never Expires so that users are not prompted to change their passwords. If you are restricting the change password function to a limited number of users, this setting is probably the best overall setting because most users are never automatically prompted to change their password.

Content Manager OnDemand for Multiplatforms

The security user exit runs the **ARSUSEC** program when a user attempts to log on to the system. A sample C program is provided in the EXITS directory. To implement your own security user exit program, add your specific code to the sample that is provided (for example, you can call another program from the **ARSUSEC** program). For more information about functions, parameters, and return codes, see the ARSCSXIT.H file. Then, compile the **ARSUSEC** program and move or copy the executable program to the BIN directory. Then, restart the library server to use the security user exit program.

The **arsuperm** (permissions exit) can be modified in the same way and needs to be placed in the /opt/IBM/ondemand/V9.5/exits directory.

Content Manager OnDemand for i server

By default, the Content Manager OnDemand for i server activates the security exit and uses IBM i security. If the security exit is not enabled, the Content Manager OnDemand user ID and password have no relationship to the IBM i user ID and password, and all of the Content Manager OnDemand system parameter settings are honored. You can enable or disable this exit at an individual instance level.

User Security Exit (ARSUSEC on z/OS only)

On z/OS, the **ARSUSEC** exit invokes the **ARSUSECZ** security exit module. The security exit allows the communication with an external security manager, such as RACF, which then determines whether the specific activity is allowed.

When you enable the exits to implement the required level or type of security, the user ID must be defined for both TSO and Content Manager OnDemand.

Figure 6-12 is an overview of the security system exits interface.

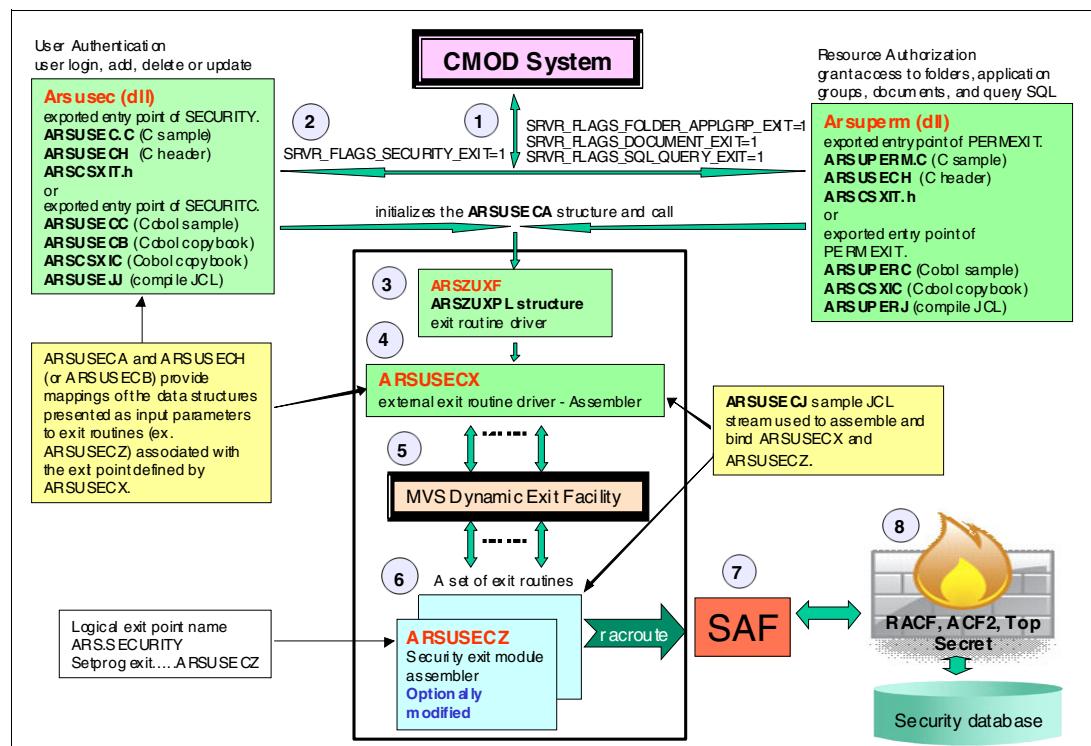


Figure 6-12 Security exits interface

With the **ARCCSXIT_SECURITY_OKAY_BUT_VALIDATE_IN_OD** return code option, a user can act on a request and then the option allows Content Manager OnDemand to perform the standard security processing. For example, do not allow a new password to match an old password in a change-password request; the password must be changed.

Table 6-2 lists the z/OS modules or executable files that ship with Content Manager OnDemand.

Table 6-2 Security exit modules

Module	Description
ARSUPERM	This c-module provides the interface between the Content Manager OnDemand system and the ARSUSECX module.
ARSUSEC	This c-module provides the interface between the Content Manager OnDemand system and the ARSUSECX module.
ARSUSECA	The mapping of the data structure that is presented to the exit routine is associated with the exit point that is defined by ARSUSEC in assembler.
ARSUSECH	The mapping of the data structure that is presented to the exit routine is associated with the exit point that is defined by ARSUSEC in C.
ARSUSECJ	This sample JCL stream is for assembling and binding ARSUSECX and ARSUSECZ .
ARSUSECX	This interface module is for the MVS Dynamic Exit Facility.
ARSUSECZ	This module is the Security Exit Module Sample.

All modules are in the SARSINST library. The sequence of this exit, using the MVS Dynamic Exit Facility, is different from the classical interface with exit modules or a security exit in an IBM CICS environment. The kernel code was updated to allow external security. The Content Manager OnDemand kernel code calls a dynamic link library (DLL) as an interface to the exit. Modules ARSUSEC and ARSUPERM are provided as C source code modules and as executable files. You do not need to change and recompile them.

The source is delivered mainly for understanding the entire security system exit. If you want to change the modules, they must be recompiled and bound as a C dynamic link library (DLL). These modules communicate with the ARSUSECX module, which is an interface to the MVS Dynamic Exit Facility. The security exit module ARSUSECZ is the delivered sample that shows how to perform security checks with a Security Exit Facility (SAF) interface. RACF is a program that uses SAF. ARSUSECH is a C source code module that passes the data structure as input for every exit (ARSUSECZ) that is provided. ARSUSEA provides the same function in assembler language.

Note: More than one security exit can be defined to the MVS Dynamic Exit Facility. For example, you can define a different security exit for each instance.

Tip: The only module that you must change is the provided source code ARSUSECZ to meet your requirements. It must be assembled and linked into a library that is accessible for the MVS Dynamic Exit Facility.

6.7.2 Security systems other than SAF (z/OS only)

The sample that is provided with the Content Manager OnDemand installation is an SAF sample. However, other installations use their own security system or use their security system as an enhancement together with the SAF environment. These systems can be accessed if they provide a correct assembler callable interface. The security exit sample code contains an example for every function. These functions can be changed or updated in the sample code.

For example, if your folder permissions are stored in an external security system without any SAF interface, this part must be updated to call this external security system.

Content Manager OnDemand SAF resource classes

You must define SAF resource classes ARS1FLDR and ARS1APGP for the folders and application group. For more information about the resource classes, see the section, “OnDemand SAF resource classes”, in the *IBM Content Manager OnDemand for z/OS - Configuration Guide*, SC19-3363.

Important: Even if the security exit can check the user ID and password against SAF or other security systems, every user must be defined in Content Manager OnDemand in every instance. You can use the **ARSXML** program to create users in batch mode, and use it as a command from the UNIX System Services command line and use a file as input.

Activating the security and permission exits (ARS.INI)

Activation of the security exit is controlled by settings in the ARS.INI file. The settings and their corresponding events are listed in Table 6-3.

Table 6-3 ARS settings and the corresponding enabled events

ARS.INI statement	Enabled event
SRVR_FLAGS_SECURITY_EXIT=1 (This setting is the default for Content Manager OnDemand for i. If you do not want to use IBM i security for the new instance, change the security setting to 0.)	Logon. Changing the password. Adding or deleting a user ID through the Content Manager OnDemand administrator interface.
SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1	Activates the folder or the application group permission.
SRVR_FLAGS_SQL_QUERY_EXIT=1	Activates the SQL query exit.
SRVR_FLAGS_DOCUMENT_EXIT=1	Activates the document permission exit.

Implementing the security exit in a z/OS environment

The module ARSUSECX interfaces with the MVS Dynamic Exit Facility:

- ▶ It defines the logical exit point name, ARS.SECURITY.
- ▶ It routes the control to a set of associated exit routines and processes the results of their execution.

Note: The sample processes the feedback of the exit one at a time, even if you are running more than one exit.

An exit routine must be eligible for execution by associating a logical exit point (ARS.SECURITY). In this example, the MVS Dynamic Exit Facility provides several methods to perform this association. You can use the PROGXX statement in Sys1.Parmlib to define exits to the Dynamic Exit Facility at IPL time (Exit statement for PROGXX).

The following example shows the exit statement for PROGXX:

```
EXIT ADD EXITNAME(ARS.SECURITY) MODNAME(ARSUSECZ)
```

In addition, you can use the following operator command to add the exit:

```
SETPROG EXIT,ADD,EXITNAME=ARS.SECURITY,MODENAME=ARSUSECZ
```

Important: The load module must be in a link pack area (LPA) or an LNLKLST dataset.

6.7.3 Unified logon exit (ARSPTGN): z/OS only

With the Content Manager OnDemand unified login exit (ARS.PTGN), you can run the Content Manager OnDemand command-line utilities (such as ARSLOAD) without requiring a specified user ID and password.

This facility to log on without specifying a password specifies a PassTicket as a password when you use a RACROUTE REQUEST=VERIFY call. Figure 6-13 shows the unified logon exit. CMOD in the figure stands for Content Manager OnDemand.

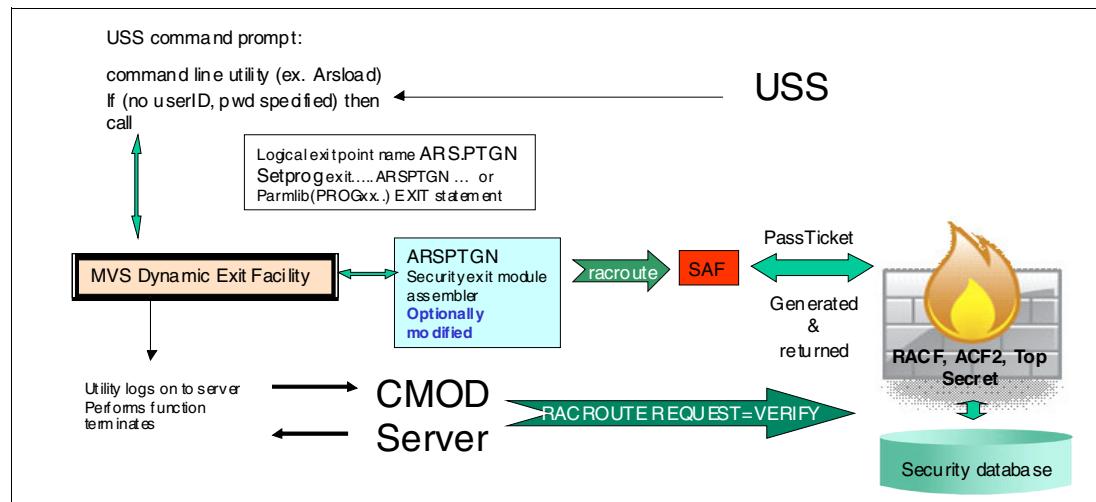


Figure 6-13 Unified logon exit

To enable PassTicket in a security manager, such as RACF, you must complete the following steps:

1. Activate the PKTDATA class.
2. Define a secured sign-on application key for each application.
3. Run SETROPTS RACLIST(PKTDATA).

6.7.4 System log user exit

Content Manager OnDemand generates messages about the various actions that occur on the system. For example, when a user logs on the system, Content Manager OnDemand generates a message that contains the date and time, the type of action, the user ID, and other information. Unless you specify otherwise, certain messages are automatically saved in the system logging facility. You can configure the system to save other messages in the system logging facility.

The system log user exit allows access to all of these messages. The exit can then use these messages for further processing. For example, an email can be generated when a load fails, or when a user's system access pattern is abnormal and requires attention. For more information about the system log, see 11.4.1, "System log exit for Multiplatforms" on page 250 and 11.4.2, "System log exit for z/OS" on page 253.

6.8 Summary

Content Manager OnDemand provides a secure environment. Security features within Content Manager OnDemand allow access control to the data and the APIs that access the data. The data itself is controlled at rest and in motion (SSL). Additional exits that are external to Content Manager OnDemand can be created that allow the creation of customized extensions to the Content Manager OnDemand internal security.



Part 2

Data indexing, loading, retrieval, and expiration

This part contains the following chapters:

- ▶ Chapter 7, “Indexing and loading” on page 161
- ▶ Chapter 8, “User clients” on page 185
- ▶ Chapter 9, “Data conversion” on page 207
- ▶ Chapter 10, “Migration and expiring data and indexes” on page 219
- ▶ Chapter 11, “Exits” on page 241



Indexing and loading

In this chapter, we describe the various indexers that are available for IBM Content Manager OnDemand (Content Manager OnDemand).

In this chapter, we cover the following topics:

- ▶ Introduction
- ▶ Getting started with PDF indexing
- ▶ Getting started with ACIF indexing
- ▶ OS/390 indexer on z/OS and AIX
- ▶ OS/400 indexer on Content Manager OnDemand on IBM i
- ▶ User exits
- ▶ Additional references

7.1 Introduction

Before documents can be loaded into Content Manager OnDemand, they must be indexed. These indexes can be created during the load process (OS/390 indexer), directly before the load process (Advanced Function Presentation (AFP) Conversion and Indexing Facility (ACIF), OS/400, XML, and Portable Document Format (PDF) indexers), or before the load process (Generic indexer). When the indexes are not created as part of the load process, they are stored in an *index file*. The index file contains the index values that are associated with the document and “pointers” to the documents. You cannot load documents into Content Manager OnDemand without index values.

The index values are text strings that occur in the documents, for example, “John Doe”, or “Account 1234”. One or more index values identify a unique document in Content Manager OnDemand.

An *indexer* extracts the index values and optionally stores them in the index file by examining the documents and copying the index values into the index file according to criteria that are specified by the user. Depending on the indexer that is used, the data and indexes are either directly loaded into Content Management OnDemand or are stored in a set of files that are then read by the load process to store the data to Content Manager OnDemand. The indexer creates the following files:

- ▶ Output file (.out file extension), which contains the documents to load
- ▶ Index file (.ind file extension), which contains the index values for the documents

The indexer might also create a resource file with a .res extension, which contains the resources that are extracted from the documents.

Operationally, the loading process **arsload** calls the indexer that is specified on the Indexer Information tab for the specified application. Depending on the indexer type, **arsload** performs one of the following tasks:

- ▶ Creates a set of files that is then loaded by the **arsload** program into the Content Manager OnDemand System
- ▶ Directly passes the indexing and document information to the **arsload** program so that they can be loaded into the Content Manager OnDemand System

On Content Manager OnDemand for i, **arsload** is embedded within the (**ADDPRTOND**) user interface. Therefore, run the Add Report to Content Manager OnDemand (**ADDPRTOND**) command instead of **ARSLOAD**.

It is possible for the indexing to complete successfully but for the load to fail. The following reasons are the most common reasons for a loading failure:

- ▶ Using insufficient system resources
- ▶ Connecting to the wrong database
- ▶ Extracting the wrong index value from the document

For information about investigating and resolving common load failures, see 18.1.2, “Indexing and loading issues” on page 379.

7.1.1 Loading and indexing files that were created on another system

Reports and documents are often created on a platform other than the platform on which the Content Manager OnDemand Instance is installed. Two main ways exist to load these reports and document files:

- ▶ Transfer the files from the remote system to the system that contains the Content Manager OnDemand instance and then index and load the documents on that system.

Many applications are available for transferring files.

For example, if your reports are generated on a z/OS system and you want to load them from a Microsoft Windows system, you can use these methods:

- On the z/OS side, use the “Download for z/OS” application to automatically download the files from the z/OS system. “Download for z/OS” is a utility that is included as part of the Print Services Facility for z/OS.
- On the receiving side (in this case Windows), you can use the Content Manager OnDemand ARSJESD utility. The ARSJESD utility runs as a service on Windows, and it runs as a daemon on other platforms.

For more information about ARSJESD, see the *IBM Content Manager OnDemand for Multiplatforms Administration Guide*, SC19-3352.

- ▶ Run the indexing and load program on the remote system. In this case, the load program sends the documents and indexes to the Content Manager OnDemand System through the TCP/IP network. To run the index and load programs on the remote system, you must copy the appropriate Content Manager OnDemand product code to that system.

You can choose to use either or both of these methods for your remote data loading.

7.1.2 Understanding input data types

It is important to know the data type of the documents that you load into Content Manager OnDemand. By *data types*, we mean document formats, such as Line Data, SCS, AFP, or PDF. In addition to knowing the data type, if you are loading line data, it has the following characteristics:

- ▶ Fixed length or variable records
- ▶ If variable, stream or 2-byte length prefix
- ▶ If stream, identify the record delimiter
- ▶ Whether carriage controls are present
- ▶ Type of carriage control, American National Standards Institute (ANSI) or machine
- ▶ Whether Table Reference Character (TRC) codes are present
- ▶ Code page of the data

Run **arsafpd** to determine the input data type of your file. Knowing the input data type determines the indexer that you can use and also helps you determine several of the indexing parameters that you need.

To run **arsafpd** from the command line, enter the following command:

```
arsafpd -s -i <input file>
```

Figure 7-1 on page 164 shows examples of running the **arsafpd** command and the output that is produced.

```

arsafpd -s -i testfile1.txt
ARS7104I Document type: LINE
ARS7114I Records appear to be delimited by hex character(s): 0x0A
ARS7115I Codepage appears to be: ASCII
ARS7110I Carriage control type appears to be: NONE
ARS7111I Pages appear to be delimited with a formfeed (0x0C). The
asciinp and asciinpe userexit might be required if using ACIF.

arsafpd -s -i testfile.afp
ARS7104I Document type: AFP
ARS7107I Group TLE structured fields were encountered

arsafpd -s -i admin.pdf
ARS7104I Document type: PDF

```

*Figure 7-1 Examples of running the **arsafpd** command and the output that is produced*

You can also run the **arsafpd** command to display the contents of an AFP document, index, or resource file. For more information about ARSAFPD, see the *Content Manager OnDemand for Multiplatforms Administration Guide*, SC19-3352.

7.1.3 Choosing an indexer

You choose the indexer to use based on multiple factors, including the data type of the documents, the platform on which you are running the indexer, and other criteria. The main factors are listed in Table 7-1. Many other factors, such as cross-platform compatibility, advanced indexing functions, and expertise, exist.

Table 7-1 Indexers that are available for use with Content Manager OnDemand

Indexer	Input data type	Available platforms	Conversion	Resource collection	Large object support	Floating triggers
Generic	All	All	No	No	No	No
ACIF	Line, AFP	All, except IBM i	Line to AFP	Yes	Yes	Yes
PDF	PDF	All, except z/OS	No	Yes	No	Yes
OS/400	Line, AFP, SCS, and SCS-Ext	IBM i	SCS to AFP	Yes	Yes	Yes
OS/390	Line, AFP	z/OS and AIX	No	Yes	Yes	Yes
XML	XML	All	No	Yes	No	No

Consider the following information about Table 7-1 on page 164:

- ▶ The Generic indexer requires the user to manually create an index file in the generic index format before the user starts the load process. The Generic indexer allows the capture of documents, index values, and resources that are identified to it. These documents, index values, and resources are then loaded into the Content Manager OnDemand archive and stored in the same manner as though they were loaded through any of the other indexers. An existing resource file can be loaded with a generic index file.

For more information about the generic index format, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

- ▶ The ACIF, PDF, XML, and OS/400 indexers all generate intermediate files. These files are then used to load the indexes and data into the Content Management OnDemand system.
- ▶ The OS/390 indexer creates the index data while it loads the indexes and data into the Content Management OnDemand system.
- ▶ *Conversion* refers to a conversion by the indexer. Other products integrate with Content Manager OnDemand that also convert data.
- ▶ Because of the architecture of PDF documents, large object support for PDF documents is not possible.
- ▶ Starting with V9.5, the PDF Indexer runs in the PASE environment on IBM i. PASE is a prerequisite on IBM i for V9.5.
- ▶ Starting with V9.5, the PDF Indexer is no longer supported on z/OS.

7.2 Getting started with PDF indexing

PDF is a standard that is specified by Adobe Systems, Incorporated, for the electronic distribution of documents. PDF files are compact. They can be distributed globally through email, the web, intranets, or CD-ROM, and viewed with Adobe Reader.

PDF is a data type or file format that is platform (hardware, operating system)-independent. A PDF file contains a complete PDF document that is composed of text, graphics, and the resources that are referenced by that document.

Two PDF file layouts are possible:

- ▶ Non-Linear (not “optimized”)

This file layout is optimized for space savings. Storing a PDF file by using a Non-Linear layout consumes less disk space than storing the same PDF file linearly. It is slower to access or display this type of layout because portions of the data that are required to assemble pages of the document are scattered throughout the PDF file, so the whole PDF file must be downloaded and accessed before the file can be displayed.

- ▶ Linear (“optimized” or “web optimized”)

In this file format, the PDF file is created in a linear (in page order) fashion. This file format allows the PDF viewer to start displaying the PDF document pages when they are downloading without waiting for the whole PDF file to be downloaded.

7.2.1 Limitations

The maximum input file size that is supported by PDF Indexer is 4 GB. The amount of data that can be processed from an input file is also limited by the amount of memory that is available on the server on which you are running the PDF Indexer. The maximum size of a single document within the input file that can be loaded into Content Manager OnDemand is 2 GB; however, we suggest that the size of a single PDF document does not exceed 50 MB.

Secure PDF documents are not supported. PDF Digital Signatures are not supported. If a PDF document contains a digital signature, after indexing, the .out file does not contain the digital signature. To load a file that contains a PDF Digital Signature, create a generic index file for it, and load the file as one document.

7.3 Performance considerations

The best performance of the PDF Indexer is on the Windows platform. For the preferred performance practices, see 13.4.1, “PDF data” on page 308.

7.3.1 PDF fonts and output file size

The fonts that are used in a PDF document are one of the factors that determines the indexing's output file size.

The base 14 Type 1 fonts

The base 14 Type 1 fonts are a core set of fonts that are always available to the Acrobat program. Because they are available on the system, they are *not* embedded in the document. Therefore, documents that are created with these fonts are more compact. The base 14 fonts are listed:

- ▶ Courier
- ▶ Courier-Bold
- ▶ Courier-BoldOblique
- ▶ Courier-Oblique
- ▶ Helvetica
- ▶ Helvetica-Bold
- ▶ Helvetica-BoldOblique
- ▶ Helvetica-Oblique
- ▶ Times-Roman
- ▶ Times-Bold
- ▶ Times-Italic
- ▶ Times-BoldItalic
- ▶ Symbol
- ▶ ZapfDingbats

Fonts that are not members of the base 14 fonts might be embedded in the document, or they might be stored in a font directory.

Images and bar code fonts are also embedded in the document.

The PDF Indexer collects resources, such as fonts and images, removes them from the document, and places them in a resource file. The number of embedded fonts in the document directly affects the size of the resource file.

We recommend that you use only the base 14 fonts when you create PDF documents. Because these fonts are not embedded in the document, documents that are created with these fonts are smaller, and the resource file is also smaller.

Accessing fonts

If a document references fonts that are not embedded and fonts that are not available on the system, the document does not display correctly in the report wizard, and the PDF Indexer cannot index it. In the report wizard, the document might display as a series of dots instead of letters; the PDF Indexer fails with the “Trigger not found” message.

If your documents contain Asian fonts, ensure that you install them when you install Adobe Acrobat.

If the fonts are not embedded in the document, use the **FONTLIB** parameter to tell the PDF Indexer the location of font files.

Listing fonts in a PDF file

If you want to know the fonts that are contained in a PDF document, a simple method within the Adobe viewer is available to list the fonts in your data.

Follow these steps to list the fonts in a PDF (for example, for Adobe Reader XI, version 11.0.3):

1. Display your PDF document in the Adobe viewer (or reader).
2. Click **File** → **Document Properties** → **Fonts**. You will see a list of fonts for the document.

The path to see the fonts might differ, depending on your viewer version.

7.3.2 Reducing output file size with PDF documents

When you index PDF data, you might be surprised by the size of the output file that the PDF Indexer creates after it indexes the data. In certain cases, the PDF file that is loaded into Content Manager OnDemand is many times larger than the source PDF file.

When the input file is indexed, it is split into multiple PDF documents. Each PDF document contains its own set of PDF structures that are required by the PDF architecture. For this reason, the multiple PDF documents that are created by the indexing can be larger in total than the original PDF document.

One way to reduce the size of the output file is using the base 14 fonts.

In addition, the following PDF parameter settings can help reduce the size of the output file:

- ▶ RESTYPE=ALL

The PDF Indexer removes fonts and images from the input file and places them into a separate resource file. Without this option, each PDF document that is created by the indexing contains its own set of duplicate resources. *Always* use this parameter.

- ▶ BOOKMARKS=NO

If a PDF document contains bookmarks, each PDF document that is created by the indexing process contains the complete set of bookmarks for the input file. Because the input file is now split into separate documents, most of these bookmarks are invalid. This option prevents the PDF Indexer from copying any bookmarks to the new PDF files.

- ▶ REMOVERES=YES

This option causes the PDF Indexer to remove unused resources and their supporting structures from the input file before the indexing occurs. Otherwise, the PDF Indexer puts unused resources (with those resources that are used) into the resource file.

7.3.3 PDF indexing: Using PDF metadata

When the PDF file is created, the user or application must place the metadata (indexes) in the PDF file. The metadata (indexes) within the document can be modified at any time after which a new copy of the document can be reloaded into Content Manager OnDemand.

Setting INDEXMODE=METADATA (for the application) causes the PDF Indexer to extract fields from the Document Information Dictionary that correspond to the specific metadata keywords (if they exist) and place the extracted values into the .ind file to load into Content Manager OnDemand. The metadata keywords are listed:

- ▶ Title
- ▶ Author
- ▶ Subject
- ▶ Creator
- ▶ Producer
- ▶ CreationDate
- ▶ ModDate
- ▶ Trapped

The main advantage of using metadata is the increased speed during the index process. The main disadvantage of using this method is that each document needs to be loaded individually; you cannot create large concatenated (multiple document) input data files.

For more information about using PDF metadata, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

7.3.4 PDF indexing: Using the report wizard (graphical indexer)

The report wizard, which is also known as the graphical indexer (technically part of the report wizard), processes PDF input files.

If you plan to use the report wizard, you *must* first install Adobe Acrobat on the Windows workstation from which you plan to run the Administrator Client. You must purchase Adobe Acrobat from Adobe.

Installation

Content Manager OnDemand provides the ARSPDF32.API file to enable PDF viewing from the client.

If you install the client after you install Adobe Acrobat, the installation program copies the application programming interface (API) file to the Acrobat plug-in directory.

If you install the client before you install Adobe Acrobat, you must copy the API file to the Acrobat plug-in directory manually.

If you upgrade to a new version of Acrobat, you must copy the API file to the new Acrobat plug-in directory.

The default location of the ARSPDF32.API file is:

C:\Program Files (x86)\IBM\OnDemand Clients\V9.5\PDF

The default Acrobat plug-in directory is C:\Program Files (x86)\Adobe\Acrobat x.y\Acrobat\plug_ins. The variables x.y represent the version of Acrobat, for example, C:\Program Files (x86)\Adobe\Acrobat 10.0\Acrobat\plug_ins.

Graphical indexer example

By using the graphical indexer, you can define triggers, fields, and indexes for PDF reports within the application component of Content Manager OnDemand in a similar way to defining them for line data. This section serves as an introduction to the PDF graphical indexer by stepping through an example of indexing a PDF document.

The example describes how to use the graphical indexer from the report wizard to create indexing information for an input file. The indexing information consists of a trigger that uniquely identifies the beginning of a document in the input file and the fields and indexes for each document. We elaborate on this example by clarifying several of the instructions, and throughout each step, we add important hints, tips, and explanations.

The process consists of these steps:

1. Start the Administrator Client and log on to a server.
2. Start the report wizard. Click the report wizard icon on the toolbar.
3. In the Sample Data window, select **PDF** from the drop-down list of data types, and then click **Select Sample Data**.
4. In the Open window, enter the name or full path name of your file in the space that is provided or use the **Browse** option to locate your PDF file.
5. Click **Open**. The graphical indexer opens the input file in the report window.

If the PDF data fails to display, or an error message, such as the message that is shown in Figure 7-2, is displayed, you must follow the steps in “Installation” on page 169 to verify that the API file is in the correct Acrobat plug-in directory.



Figure 7-2 Error message if PDF does not display

6. Press F1 to open the main help topic for the report window.

The main help topic contains general information about the report window and links to other topics that describe how to add triggers, fields, and indexes. For example, to get help to define a trigger, click **Adding a trigger (PDF)**. You can also use the context help tool to display information about the icons on the toolbar.

7. Close any open help topics and return to the report window.

8. To define a trigger, complete the following steps:

- a. Find a text string that uniquely identifies the beginning of a document, for example, Account Number, Invoice Number, Customer Name.

Note: To create trigger values in hexadecimal, select the **Output Hexadecimal Strings** check box in the Indexer Properties window before you define a trigger.

- b. By using the mouse, draw a box around the text string. Start just outside of the upper-left corner of the string. Click and then drag the mouse toward the lower-right corner of the string. As you drag the mouse, the graphical indexer uses a dotted line to draw a box. After you enclose the text string inside a box, release the mouse. The graphical indexer highlights the text string inside the box. If the string is not highlighted, try again and increase the box's size.

Important: Size the box that you created around the text string, which you are trying to collect, as large as possible to ensure that the field is collected at load time.

Figure 7-3 on page 171 shows an example of a box that is intended to capture the text string Content. You can see that the box is much larger than the text string, and it overlaps onto text that we do not want to collect. However, notice the Add a Trigger box that is displayed; only the string Content is shown in the Value entry field, which means that only the string Content is *fully* encapsulated in the box. Overlapping other text might seem like an unnecessary precaution. However, when we are capturing data with the PDF graphical indexer, it is an excellent way to ensure that we encapsulated *all* of the text string that we must capture.

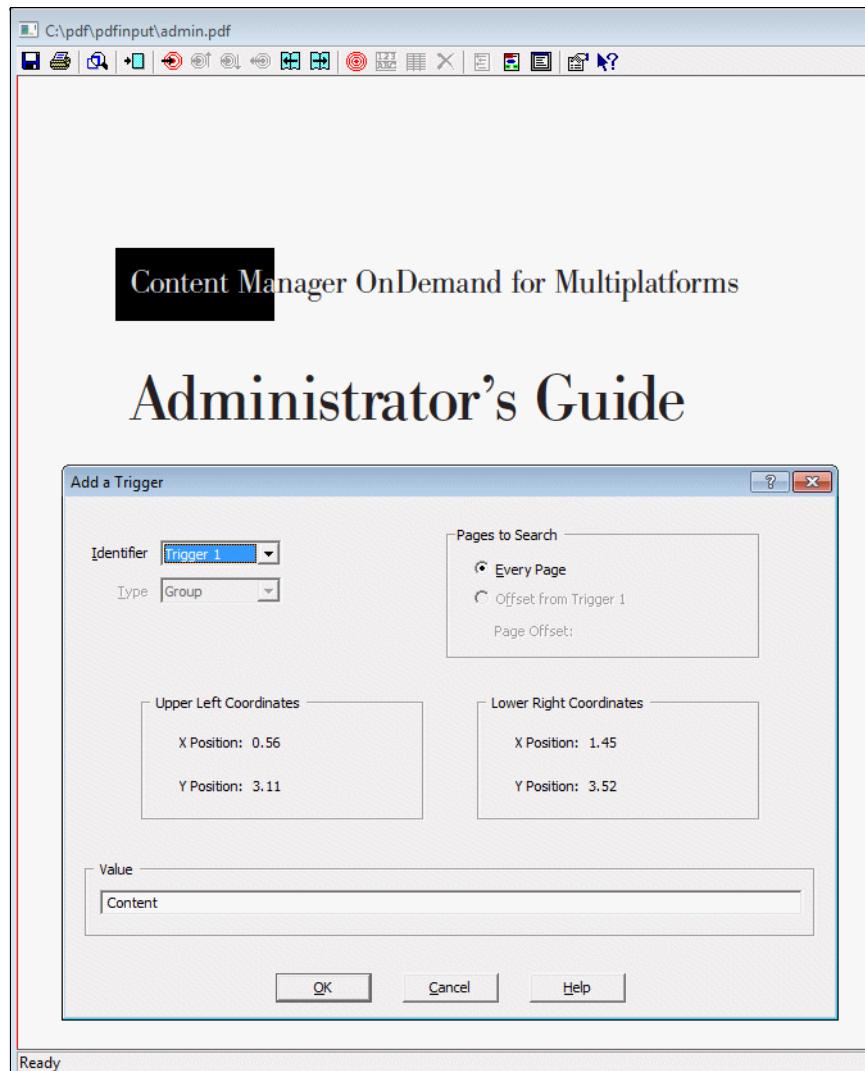


Figure 7-3 Capturing text with the PDF graphical indexer

- c. Click the Define a Trigger icon on the toolbar.
- d. In the Add a Trigger window (Figure 7-3), verify the attributes of the trigger by confirming that the text string in the Value field for Trigger 1 is correct. For Trigger 1, you cannot specify any options or values. For other triggers, click **Help** for assistance with the other options and values. Click **OK** to define the trigger.
- e. Follow these steps to verify that the trigger uniquely identifies the beginning of a document:
 - i. On the toolbar, click the fourth icon from the right to place the report window in display mode.
 - ii. Click the **Select** tool.
 - iii. In the Select window, under Triggers, double-click the trigger. The graphical indexer highlights the text string in the current document. Double-click the trigger again. The graphical indexer highlights the text string on the first page of the next document.
 - iv. Use the Select window to move forward to the first page of each document and return to the first document in the input file.

- f. On the toolbar, click the fourth icon from the right to place the report window back into add mode.
9. Define a field and an index:
- a. Find a text string that can be used to identify the location of the field. The text string needs to contain a sample index value. For example, if you want to extract account number values from the input file, find where the account number is printed on the page.
 - b. By using the mouse, draw a box around the text string. Start just outside of the upper-left corner of the string. Click and then drag the mouse toward the lower-right corner of the string. As you drag the mouse, the graphical indexer uses a dotted line to draw a box. After you enclose the text string inside of a box, release the mouse. The graphical indexer highlights the text string inside the box.

Important: Use the same principles for collecting fields as collecting the trigger text string in step 8b on page 170. If the fields that must be collected are close together, overlap them with adjacent fields to ensure that the box is as large as possible and to ensure that the data is collected at load time.

- c. Click the Define a Field icon on the toolbar.
 - d. In the Add a Field window, complete the following steps:
 - i. On the Field Information tab, verify the attributes of the Index field. For example, the text string that you selected in the report window is displayed under Reference String and the trigger identifies the trigger on which the field is based. Click **Help** for assistance with the options and values that you can specify.
 - ii. On the Database Field Attributes tab, verify the attributes of the database field. In the Database Field Name field, enter the name of the application group field into which you want Content Manager OnDemand to store the index value. In the Folder Field Name field, enter the name of the folder field to display in the client search window. Click **Help** for assistance with the other options and values that you can specify.
 - iii. Click **OK** to define the field and index.
 - e. To verify the locations of the fields, complete the following steps:
 - i. Place the report window into display mode. Blue boxes are drawn around the fields.
 - ii. Click the **Select** tool.
 - iii. In the Select window, under Fields, double-click **Field 1**. The graphical indexer highlights the text string in the current document. Double-click **Field 1** again. The graphical indexer moves to the next document and highlights the text string.
 - iv. Use the Select window to move forward to each document and display the field. Then, return to the first document in the input file.
 - f. Place the report window back into add mode.
10. Click **Create Indexer Parameters and Fields Report** to create the indexer parameter report that the PDF Indexer uses to process the input files that you load into the application. At a minimum, you must have one trigger, one field, and one index. For more information about the indexing parameters, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.
11. After you define all of the triggers, fields, and indexes, press Esc to close the report window.

12. Click **Yes** to save the changes to the indexer parameters.

13. In the Sample Data window, click **Next** to continue with the report wizard.

7.3.5 PDF indexing: Using internal indexes (Page Piece Dictionary)

When the PDF document is created, the user or application must insert indexes into the Page Piece Dictionary. For Content Manager OnDemand, the Page Piece Dictionary must be named “IBM-ODIndexes” to allow the PDF Indexer to find the Page Piece Dictionary and collect the index values.

Setting `INDEXMODE=INTERNAL` (for the application) causes the PDF Indexer to segment the input file into the individual documents, gather the various PDF resources (fonts, images, and forms), and then load the PDF indexes, documents, and resources into Content Manager OnDemand.

The use of internal indexes offers multiple advantages:

- ▶ Fast indexing: A single PDF file can contain many PDF documents. Extracting the indexes for these documents is now fast because Content Manager OnDemand now scans the documents and reads the index values directly from the Page Piece Dictionary. (No search exists for the indexes within the document data.)
- ▶ Different formats can exist in a single PDF input file: This flexibility is possible if the indexes are similar because only the index is read and processed by Content Manager OnDemand.
- ▶ The indexed PDFs can be either static or dynamic: Static PDF forms render once and are displayed on the client in Adobe Acrobat or Adobe Reader. Static PDF forms are not re-rendered in response to user interactions. Dynamic PDF forms render on the client in Adobe Reader and, depending on the user interactions, can re-render on the client several times. Re-rendering causes the content of the form (all objects, including text and image) to change.

Both the static and dynamic PDFs can be indexed because the PDF Indexer is only looking at the Page Piece Dictionary. The PDF document data is not examined or processed.

For more information about using internal indexes (Page Piece Dictionary), see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

7.4 Getting started with ACIF indexing

The AFP Conversion and Indexing Facility (ACIF) consists of three separate but related functions. ACIF can perform the following tasks:

- ▶ Convert line data to AFP.
- ▶ Index line or AFP data.
- ▶ Collect resources.

ACIF accepts either line data or AFP as input and can produce three output files:

- ▶ The output file, which is called the “out” file, is either line data or AFP.
- ▶ The index file, which is called the “ind” file, is an AFP file.
- ▶ The resource file, which is called the “res” file, is an AFP file.

Three “modes” of running ACIF are available:

- ▶ Mode one: Line data input to ACIF creates line data output:
 - Specify the ACIF parameter CONVERT=NO.
 - ACIF does not create a resource file.
 - Files produced: .out and .ind.
- ▶ Mode two: Line data input to ACIF creates AFP output:
 - Specify the ACIF parameter CONVERT=YES.
 - ACIF creates an AFP resource file.
 - Files produced: .out, .ind, and .res.
- ▶ Mode three: AFP input to ACIF creates AFP output:
 - Specify the ACIF parameter CONVERT=YES.
 - ACIF creates an AFP resource file.
 - Files produced: .out, .ind, and .res.

A subset of the second mode is mixed mode input (line data records mixed with AFP records). In this case, ACIF creates AFP output:

- ▶ Specify the ACIF parameter CONVERT=YES.
- ▶ ACIF creates an AFP resource file.
- ▶ Files produced: .out, .ind, and .res.

Types of ACIF parameters

Because ACIF has so much functionality, it has many parameters. Four logical sets of ACIF parameters are available:

- ▶ ACIF parameters that describe the format of the data: **CC**, **CCTYPE**, **TRC**, **FILEFORMAT**, and **CPGID**
- ▶ ACIF parameters for line data to AFP conversion: **CONVERT**, **MCF2REF** (we recommend coded font (**CF** parameter) instead of code page character set (**CPCS** parameter)), **IMAGEOUT** (we recommend **ASIS** parameter instead of Image Object Content Architecture (IOCA)), **FORMDEF**, and **PAGEDEF**
- ▶ ACIF parameters for indexing: **TRIGGER**, **FIELD**, **INDEX**, **INDEXOBJ**, and **INDEXSTARTBY**
- ▶ ACIF parameters for collecting resources: **RESTYPE** and **EXTENSIONS=RESORDER**

For a description of the parameters, see the section “ACIF reference” in *IBM Content Manager OnDemand for Multiplatforms Indexing Reference*, SC19-3354, or “ACIF reference” in *IBM Content Manager OnDemand for z/OS Indexing Reference*, SC19-3368.

Tools for working with ACIF

Consider the use of the following tools when you work with ACIF:

- ▶ The Administrator line data graphical indexer
- ▶ A hexadecimal editor to display the input file
- ▶ The **arsafpd** utility, run with the **-d** and **-t** options

The **arsafpd** utility can display the .out file (if it is AFP), .ind file, and .res file that are created by ACIF.

7.4.1 Understanding the input data

On every platform except z/OS, the **FILEFORMAT** parameter is used to describe the format of the input data. Before setting the **FILEFORMAT** parameter, it is important to understand the difference between the carriage control and the delimiter:

- ▶ The delimiter separates the records. The most common delimiters are x'0A' and x'0D0A'.
- ▶ The carriage control, if it exists, is the first byte of each record. The carriage control follows the delimiter, except at the beginning of the file, where the carriage control is the first byte. (Therefore, to search in a hexadecimal editor for the beginning of the next page of a file that uses x'0A' as the delimiter, search for x'0AF1' or x'0A31'.)

FILEFORMAT parameter

For AFP data, the **FILEFORMAT** parameter is not needed, unless the file is AFP in record format. For a description of record format, see “AFP Structured Fields” on page 176.

The **FILEFORMAT** parameter has the following values:

- ▶ record,n:
 - For example: FILEFORMAT=record,100.
 - Fixed-length line data.
 - This type of file has no delimiter.
- ▶ stream:
 - For example: fileformat = stream, (newline=X'0A') or (newline=X'0D0A').
 - For variable record files that are created on UNIX platforms.
 - Specify the delimiter in the **FILEFORMAT** parameter.
- ▶ record:
 - For example: FILEFORMAT=record.
 - Each record has a 2-byte prefix, which contains the length of the record. This length is exclusive, which means that it does not include the length of the 2-byte prefix itself. A download for z/OS adds this prefix when it downloads files.
 - This type of file has no delimiter.

Carriage controls

It is important to set the ACIF parameters **CC** and **CCTYPE** correctly. Table 7-2 describes the ANSI carriage controls. The encoding columns show what you see if you look at the document in a hexadecimal editor.

Table 7-2 ANSI carriage controls

Carriage control	Description	Encoding in ASCII	Encoding in EBCDIC
1	New page	x'31'	x'F1'
<space>	Space one line	x'20'	x'40'
0	Space two lines	x'30'	x'F0'
-	Space three lines	x'2D'	x'60'
+	Suppress space	x'2B'	x'8F'

Machine carriage controls

Machine carriage controls are in data that is created on z/OS.

Because machine carriage controls are binary values, if a file contains them, it must always be transferred as binary. Machine carriage controls cannot be converted to ASCII. For a list of machine carriage control values, see the following website:

<http://ibm.co/1M2ZtSG>

AFP Structured Fields

AFP, which is also called Mixed Object Document Content Architecture (MODCA), is a printing architecture that was designed and created by IBM. The beginning of each AFP record is called the *AFP Structured Field Introducer*. The following sample shows an example and description of an AFP Structured Field Introducer (which is shown in the hexadecimal):

5A 00 10 D3 A8 A8 00 00 00

- ▶ The first byte is always x'5A'.
- ▶ The second and third bytes are the length (maximum length of 32767).
- ▶ The fourth byte is always x'D3'.
- ▶ The fourth, fifth, and sixth bytes are the Structured Field Identifier, for example, x'D3A8A8' or x'D3A8AF'.
- ▶ The seventh byte is the flag byte. The last two bytes are reserved and usually zeros.
- ▶ The information that follows the reserved bytes depends on the Structured Field.
- ▶ The length does not include the x'5A'.

For more information, see the *Mixed Object Document Content Architecture (MO:DCA) Reference*, AFPC-0004-08, at the following website:

<http://afpcinc.org/afp-publications/>

The following two examples in hexadecimal of the AFP Structured Field Introducer show the most common Structured Fields that you might see at the beginning of an AFP file:

5A 00 10 D3 A8 A8 00 00 00 Begin Document (BDT)

5A 00 5B D3 A8 C6 00 00 00 Begin Resource Group (BRG)

An AFP Structured Field can begin with the 2-byte length prefix (which is called *record format*):

00 11 5A 00 10 D3 A8 A8 00 00 00

The length in the 2-byte prefix is one greater than the length in the Structured Field because the 2-byte prefix includes the x'5A', but it does not include itself.

When you work with ACIF, it is important to know the format of the data. Use the **arsafpd** utility or look at the input in a hex editor to be sure.

7.4.2 The index file

ACIF creates the index file. It contains the index values that are extracted from the document, and also the offsets and lengths of the documents in the .out file.

The index values in the index file become the values that display in the Content Manager OnDemand Search Results window. The indexes are used to retrieve the document, which is why the index file is so important, and why no data can be loaded without indexes. Usually, the index file is created and used to load the documents into Content Manager OnDemand and you never see it. However, it might be useful to look at the index file. This section describes the format and content of the index file.

Run **arsafpd** to display an index file. The first Structured Field in the index file is a Begin Document Index (BDI), which contains the code page of the index names and values. Most of the file consists of the two AFP Structured Fields: Index Element (IEL) and Tag Logical Element (TLE). Two kinds of IELs exist: Page Group and Page. The index file must contain Page Group IELs for **arsload** to load the data.

A Page Group IEL is identified by the text “Begin Page Group Reference” in the **arsafpd** output. Each Page Group IEL indicates where the group starts and its length in bytes. Example 7-1 shows part of a Page Group IEL.

Example 7-1 Part of a Page Group IEL

2	IEL Index Element	005D D3B2A7
	IEL Object Byte Extent Triplet (57)	
	IEL Extent = 1614 (64E) <- LENGTH OF GROUP	
	IEL Object Byte Offset Triplet (2D)	
	IEL byte offset = 201 (C9) <- WHERE IT STARTS IN THE .OUT FILE	
	IEL Object Structured Field Extent Triplet (59)	
	IEL Extent = 18 (12)	
	IEL Object Structured Field Offset Triplet (58)	
	IEL Offset = 1 (1)	
	IEL Medium Map Page Number Triplet (56)	
	IEL sequence number of page = 1 (1)	
	IEL Fully Qualified Name Triplet (02)	
	IEL OD Begin Page Group Reference <- PAGE GROUP IEL	
	IEL Name = 'Smith Cyclery Co' 00000001'	

If you look at offset 201 in the .out file, you find a BNG Structured Field (if the .out file is AFP), which indicates the start of a document.

You might see Page IELs in the index file. These Page IELs are created by setting the ACIF parameter INDEXOBJ=ALL. They are needed (and are required) only if the document is being loaded as large object. Example 7-2 shows part of a Page IEL.

Example 7-2 Part of a Page IEL

7	IEL Index Element	0044 D3B2A7
	IEL Object Byte Extent Triplet (57)	
	IEL Extent = 1342 (53E) <- LENGTH OF PAGE	
	IEL Object Byte Offset Triplet (2D)	
	IEL byte offset = 456 (1C8) <- WHERE IT STARTS IN THE .OUT FILE	
	IEL Object Structured Field Extent Triplet (59)	
	IEL Extent = 11 (B)	
	IEL Object Structured Field Offset Triplet (58)	
	IEL Offset = 7 (7)	

```
IEL Medium Map Page Number Triplet (56)
IEL sequence number of page = 1 (1)
IEL Fully Qualified Name Triplet (02)
IEL 87 Begin Page Reference      <- PAGE IEL
IEL Name = '00000001'
```

Example 7-3 shows a Tag Logical Element (TLE) that contains index information.

Example 7-3 TLE that contains index information

3 TLE Tag Logical Element	0032 D3A090
TLE Fully Qualified Name Triplet (02)	
TLE OB Attribute Name	
TLE Name = 'NAME'	
TLE Attribute Value Triplet (36)	
TLE Value = 'Smith Cyclery Co'	'
TLE Attribute Qualifier Triplet (80)	
TLE sequence number = 0 (0)	

Summary of index file information

The index file information is summarized:

- ▶ **arsload** uses the code page value in the BDI to convert the index names and values to the code page of the database. For example, the index names and values are in EBCDIC, but the database might be in ASCII.
- ▶ TLEs contain the index values that display in the Search Results window.
- ▶ Group IELs contain the offset of where the group starts in the .out file and the length of each group.
- ▶ All of this information is loaded into Content Manager OnDemand tables, and the index file is discarded.

7.4.3 Fully composed AFP input

ACIF can process an input file in AFP format that contains TLEs and BNG/ENG pairs. This data is called *fully composed AFP*.

Example 7-4 shows a portion of the **arsafpd** output of a fully composed AFP file in the correct format to load into Content Manager OnDemand.

Example 7-4 Portion of the arsafpd output of a fully composed AFP file

1 BDT Begin Document	
2 BNG Begin Named Page Group 00000001	
3 TLE Tag Logical Element	
4 TLE Tag Logical Element	
5 TLE Tag Logical Element	
6 TLE Tag Logical Element	
7 IMM Invoke Medium Map ABBB	
8 BPG Begin Page 00000001	
9 BAG Begin Active Environment Group	
10 MCF2 Map Coded Font2	
11 NOP No Operation	
12 PGD Page Descriptor	
13 PTD2 Presentation Text Desc2	

```

14      EAG End Active Environment Group
15      BCT Begin Composed-Text Block
16          PTX Presentation Text Data
17      ECT End Composed-Text Block
18      EPG End Page
19      ENG End Named Group
20      BNG Begin Named Page Group 00000002
...
4590    ENG End Named Group
4591 EDT End Document

```

Each group is surrounded by BNG/ENG Structured Fields, and each group contains TLE Structured Fields that occur after the BNG but before the BPG.

When an input file contains TLE Structured Fields, do not specify indexing parameters, such as **TRIGGER**, **FIELD**, or **INDEX**. They are not needed because the file already contains index information.

ACIF processes a file that contains TLE Structured Fields in the following way:

1. For every BNG in the input, ACIF creates a group IEL Structured Field in the index file.
2. ACIF makes a copy of the TLE Structured Fields from the input and places them into the index file. The original TLE Structured Fields are also placed into the output file.

If the input file does not contain the correct number of TLEs in each group, ACIF might complete, but **arsload** might fail with the following message:

“x fields submitted, n expected”

The *n* is the number of fields that are defined to Content Manager OnDemand.

After ACIF processes an input AFP file, the output file might be larger than the input file, even if the input was an AFP file. The answer is because ACIF changes the AFP, “improves it”, and usually increases the file size. The following changes are made to the AFP:

- ▶ Creating or adding comments to the BDT Structured Field
- ▶ Creating or adding group names to the BNG - ENG Structured Fields
- ▶ Changing obsolete Structured Fields to current Structured Fields (for example, MCF1 to MCF2, or PTD1 to PTD2)

7.5 OS/390 indexer on z/OS and AIX

The OS/390 indexer is supported on both the z/OS and AIX implementations of Content Manager OnDemand. The indexing parameters are the same for both implementations. If you are migrating from z/OS to AIX, or from AIX to z/OS, you can continue to use the OS/390 indexer and not change your indexing parameters.

You can use the OS/390 indexer to extract index data from line data and AFP reports. In addition, other data types, such as TIFF images, can be captured by using the ANYSTORE exit (ANYEXIT is described in 11.3, “OS/390 indexer exits” on page 248).

The OS/390 indexer is a single pass indexer. (It does not create an intermediate file.) It therefore provides better performance than ACIF. The COBOL Runtime Library is required on AIX to run the OS/390 indexer, and it is included in the Content Manager OnDemand Multiplatform software.

The OS/390 indexer is enhanced to allow the storage of documents (or large object segments) that exceed 2 GB. A report might contain multiple documents (or large object segments), each of which exceeds 2 GB. This enhancement does not affect the limitations that are imposed by other indexers. The limitations on the document size are based on the available hardware and any other limitations that are placed on the operating environment.

For more information about the use of the OS/390 indexer, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

7.6 OS/400 indexer on Content Manager OnDemand on IBM i

The OS/400 indexer is a powerful tool to index the print data streams of IBM i application programs. Supported data streams include SCS, AFP, and the less common SCS-Extended and Line Data.

The OS/400 indexer provides three major functions:

- ▶ Print data stream processing: The OS/400 indexer processes the output print data streams of application programs, for example, SCS, AFP, and Line Data reports. The output can be viewed, printed, and archived by Content Manager OnDemand.
- ▶ Sophisticated indexing functions: The OS/400 indexer can logically divide reports into individual items, such as statements, policies, and bills. You can define up to 32 index fields for each item in a report if you are running a Content Manager OnDemand server version that is earlier than version 9.0.0.1. Beginning at version 9.0.0.1 of the server, 128 index fields can be defined.
- ▶ AFP resource collection: For AFP spooled files, the OS/400 indexer determines the resources that are necessary to view, print, and archive the print data stream and collect the resources (except fonts, which are not stored but are mapped by the client during display). Resources allow users to view the report as it displayed in the original printed version, regardless of when or where the report was created.

The OS/400 indexer supports many advanced features:

- ▶ Multi-key indexes
- ▶ Spool File Archive compatibility
- ▶ Start Indexing on Page
- ▶ Translate Print Control
- ▶ AFP support with or without TLEs
- ▶ Large object support

The OS/400 indexer processes three input sources:

- ▶ Indexing parameters that specify how the data needs to be indexed. The indexing parameters are created when you define a Content Manager OnDemand application.
- ▶ AFP resources that are required to view and print the data if the application created an AFP print data stream.
- ▶ The print data stream, which can be in a spooled file (all data types) or in a physical file (Line Data or SCS data that was converted to Line Data with First Character Forms Control (FCFC) characters in column one of the data).

The output of the OS/400 indexer consists of an output file that contains the text of the spooled file and an index file that contains the index values that are extracted from the spooled file. Also, for AFP, the output of the OS/400 indexer contains a resource file that contains the AFP resources that are used by the spooled file (except for fonts, which are not stored but are mapped by Content Manager OnDemand Client during display). To create a resource file, the OS/400 indexer must have access to the resources that are required by the input data stream. Content Manager OnDemand stores the resources and then later retrieves the resources that are associated with a specific document when a user selects the document for viewing.

The OS/400 indexer indexes input data based on the organization of the data:

- ▶ Document organization. For reports that are made up of logical items, such as statements, policies, and invoices, the OS/400 indexer can generate index data for each logical item in the report.
- ▶ Report organization. For reports that contain lines of detail with sorted values on each page, such as a transaction log or general ledger, the OS/400 indexer can divide the report into sets of pages and generate index data for each set of pages.

Before you can index a report with the OS/400 indexer, you must create a set of *indexing parameters*. The indexing parameters describe the physical characteristics of the input data, identify where in the data stream the OS/400 indexer can locate index data, and provide other directives to the OS/400 indexer.

Indexing parameters include information that allows the OS/400 indexer to identify key items in the print data stream, *tag* these items, and create *index elements* that point to the tagged items. The OS/400 indexer uses the tag and index data for efficient and structured search and retrieval. You specify the index information that allows the OS/400 indexer to segment the data stream into individual items called *groups*. A group is a collection of one or more pages. You define the bounds of the collection, for example, a bank statement, insurance policy, phone bill, or other logical segment of a report file. A group can also represent a specific number of pages in a report. For example, you might decide to segment a 10,000 page report into groups of 100 pages. The OS/400 indexer creates indexes for each group. Groups are determined when the value of an index changes (for example, account number) or when the maximum number of pages for a group is reached.

Figure 7-4 on page 182 illustrates the data indexing and flow control for OS/400 indexer. For more information about the OS/400 Indexer, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

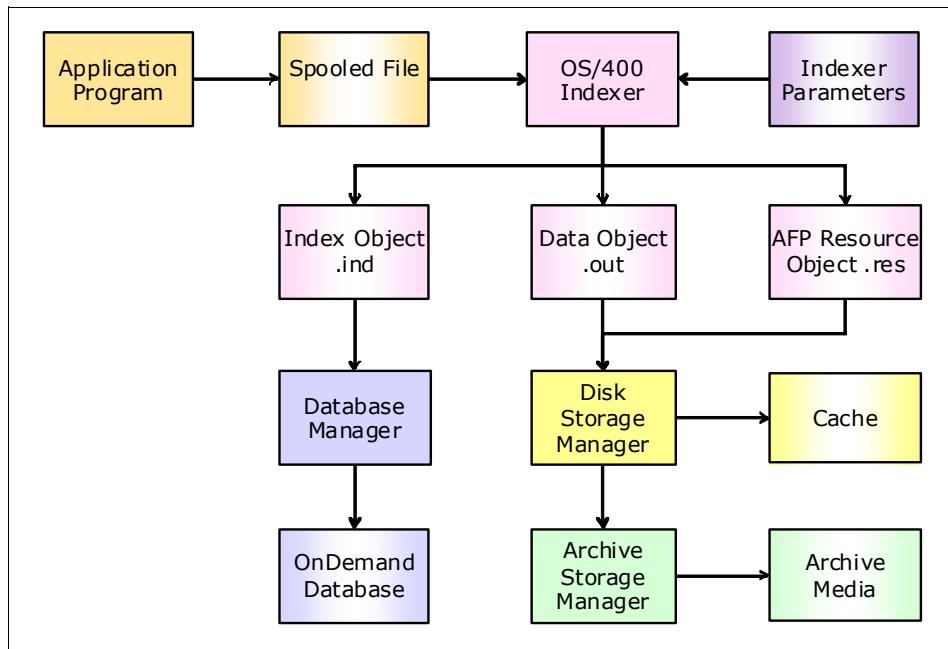


Figure 7-4 Data indexing and flow control for the OS/400 indexer

7.7 Getting started with XML Indexing

The XML indexer enables the high-volume archiving of XML data in a scalable and extensible manner.

The XML indexer was developed to support the growing need to efficiently and effectively store large quantities of XML data, for example:

- ▶ The European Union's implementation of a Single Euro Payments Area (SEPA). SEPA replaced the existing domestic retail credit transfers and direct debits with standardized European payments that are based on Extensible Markup Language (XML) International Organization for Standardization (ISO) 20022 messages. ISO 20022 provides a more efficient way of developing and implementing messaging standards that financial institutions and clients use to exchange massive amounts of transactional information.
- ▶ Other XML standards exist and continued to be developed, such as ACORD (Insurance industry), AgXML (Agriculture), and Health Level Seven (Health industry).
- ▶ XML document formats were developed, such as Office Open XML (OOXML) and Open Document (OASIS).

With XML indexing, you can automatically batch index and archive XML transactional messages and statements into the Content Manager OnDemand repository. Documents are identified and extracted during indexing. Resources are extracted, and, together with the data, compressed and archived. Multiple stylesheets can be specified to meet device and accessibility requirements.

XML steeliest (resource) archiving is critical. Content Manager OnDemand optimizes the storage of XML data by storing only a single version of a resource and then associating it with all of the archived documents. Document resources can be automatically collected and managed.

XML data is loaded into Content Manager OnDemand by using the `arsload` command. For example, the following statement loads the `bamboo.in` file and its `.res` file (if found):

```
arsload -I localhost -u userName -p load.stack -g ci_stmts bamboo,in
```

The XML indexer uses the “Generic XML Index File Format” (GXIFF). The GXIFF format is functionally similar to the Generic Index File Format in that it allows the loading of any type of data into Content Manager OnDemand.

For more information about using the XML indexer, see *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354.

7.8 User exits

A *user exit* is a point during processing where control is handed from the indexer program to a user-written program. After the user-written program finishes, the control is handed back to the indexer program.

The ACIF indexer and the OS/390 indexer support multiple user exits. The OS/400, PDF, XML, and Generic indexers do not support any user exits.

For a description of the ACIF user exits in detail, see 11.2, “ACIF exits” on page 242.

For a description of the OS/390 indexer user exits, see 11.3, “OS/390 indexer exits” on page 248.

7.9 Additional references

For more information, see the following IBM developerWorks® articles:

- ▶ *Creating PDF Indexing Parameters Using Floating Triggers:*
<http://ibm.co/1FHsXDq>
- ▶ *Understanding the ACIF Input Exit for DB2 Content Manager OnDemand:*
<http://ibm.co/1UUcCT0>



User clients

In this chapter, we provide an overview of the clients that are available for IBM Content Manager OnDemand (Content Manager OnDemand), including the various web client offerings that are based on the Content Manager OnDemand Web Enablement Kit (ODWEK). We describe the differences between web and Windows clients and their viewing options.

In the later sections, we focus on the integration and application programming interface (API) client options of Content Manager OnDemand, such as the ODWEK API, the Content Management Interoperability Services (CMIS) web services, the mid-server SAPI, and integration with other IBM Enterprise Content Manager products, such as IBM Information Integrator and IBM FileNet P8. We describe how to use the existing API to build your own web client interface for Content Manager OnDemand.

In this chapter, we cover the following topics:

- ▶ Choosing the correct client for your implementation
- ▶ Content Manager OnDemand Client options
- ▶ Client API overview

8.1 Choosing the correct client for your implementation

Customers are faced with challenges in choosing the interface to Content Manager OnDemand that makes the most sense for their implementation. Content Manager OnDemand has many different user interfaces. Many aspects come into play when you consider the best design for access to Content Manager OnDemand to meet all of your requirements in the most cost-effective manner. Licensing costs, hardware costs, performance, and maintainability are just a few considerations, but the most important requirement is meeting the business needs for many different user types.

The Content Manager OnDemand Client choices enable the product to meet the ever-changing world of information technology and the way content is delivered. For example, delivering documents that are stored in Content Manager OnDemand to a mobile device was not relevant a few years ago. However, it is an important consideration for enterprise content delivery today. Technology drives change with current Content Manager OnDemand customers, and IBM delivers options to meet current and future business requirements. A customer's goal is to use a single user interface for access to all of its Enterprise Content Management content. IBM met that goal with the IBM Content Navigator user interface, but IBM continues to retain multiple Content Manager OnDemand Client interfaces to meet the various needs of its customers.

When you choose the correct client for your implementation of Content Manager OnDemand, two primary considerations are the client functionality and the client architecture.

Concerning the client functionality, the most powerful client is the Microsoft Windows client. All other clients contain only a subset of the features of the Windows client. The most prominent difference is the viewer capability.

Determine whether your users require functionality that is specific to the Windows client only. If not, see the range of viewer options that are described in 8.1.1, “Viewer options” on page 186, which compares the different viewers across the various client options.

8.1.1 Viewer options

Different viewer options for the data that is stored within a Content Manager OnDemand system exist. The following general types of viewers are available:

- ▶ The viewing capabilities that are provided by the Windows client.
- ▶ The web viewers that are shipped with ODWEK.
- ▶ Generic web viewers that are available in Content Navigator or other third-party web viewers. The built-in viewers of Content Navigator are described in 8.2.1, “IBM Content Navigator” on page 193.
- ▶ Conversion and transformation services that are started by ODWEK.
- ▶ External applications that are opened according to their associated document types (for example, Microsoft Word for .doc or .docx files).
- ▶ Special client applications, such as the CICS client, the Structured APIs, or Java API access.

The content that is displayed by certain viewers can be changed by either transforms (ODWEK) or exits. For more information about exits, see Chapter 11, “Exits” on page 241.

Windows client viewers

The Content Manager OnDemand Windows client contains native capabilities for viewing typical archive data types:

- ▶ Line Data and SCS
- ▶ AFP
- ▶ Images

The Windows client reflects the richest set of capabilities in terms of viewing these data types. Because it directly communicates with the Content Manager OnDemand server, we reference the Windows client for all of its features that relate to document display.

The Line Data viewer of the Windows client is the most sophisticated viewer that is available for Content Manager OnDemand from the selection of readily available viewers.

The viewing of these primary data types happens within the same application. The Windows client provides other features, such as thumbnails, and configurable and saveable views.

The Content Manager OnDemand Windows client also contains other capabilities for viewing archive data types, such as Portable Document Format (*PDF*) and *User-Defined*.

Starting with Content Manager OnDemand version 9.5, for both DocType=PDF and user-defined PDF, the Windows Client will attempt to view a PDF document with Adobe Acrobat, if it is installed. If Adobe Acrobat is not installed, for DocType=PDF, Adobe Acrobat Reader will be used instead when the PDF document is viewed.

Before Content Manager OnDemand version 9.5, PDF documents can be viewed by the Windows client in two ways:

- ▶ If they are configured in the application as data type “PDF”, the rich feature set of the AFP and Line Data viewer applies, but Adobe Acrobat Professional is required.
- ▶ If the data type is configured as “User Defined” and “.pdf” as the extension, the documents are started externally. Therefore, you can view the documents with the no-charge Adobe Acrobat viewer or any other installed PDF viewer.

Any data type can be specified as “User Defined”, for example, Word documents (.docx). User-defined data is viewed by invoking its associated application.

Web-based viewing options

The web-based viewing options for Content Manager OnDemand are provided primarily by ODWEK. ODWEK includes different viewers that are dedicated to Content Manager OnDemand documents that can use Content Manager OnDemand functions, such as the segment-wise retrieval of large objects or annotations. These viewers are used in web applications, such as Content Navigator or any other custom-developed web client:

- ▶ Line Data applet
- ▶ Browser plug-in for image viewing
- ▶ AFP browser plug-in
- ▶ AFP Transforms
- ▶ Generic Transforms

Detailed information about ODWEK’s viewers and transforms is in *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646. Only a brief overview is provided in this chapter.

The *line data applet* is a Java applet that is provided by ODWEK. It is similar to the line data viewing capabilities of the Windows client, but it does not contain all of the parallel functionality for viewing line data within the Windows client. For example, the applet does not support saving and selecting custom views.

The *plug-ins for AFP and images* are shipped as setup packages, which must be installed on the user's computer. The plug-ins integrate themselves with Mozilla Firefox browsers and Microsoft Internet Explorer. The AFP plug-in provides similar viewing capabilities to the Windows client.

The image plug-in can view image files, with the added benefit of displaying TIFF images (which current web browsers usually cannot display).

Conversions and transforms

In addition to the viewers, ODWEK uses conversion or transformation engines, which convert the document into another data type. ODWEK allows the integration of AFP Transform components for converting AFP into HTML or PDF documents, and it provides a generic transform interface, which can be used to plug in any conversion or transformation engine.

The transforms apply only to documents that are served by ODWEK. They are available to web clients that are based on ODWEK (such as Content Navigator) and to any other application that is written by using the ODWEK Java API. They are not available on the Windows client.

Web viewing considerations

When you choose a viewer strategy in web clients, it is important to know the differences among the viewer architectures:

- ▶ Java applet viewers, such as the line data applet or Content Navigator's generic applet viewer, are downloaded automatically to the user's computer and run within the browser. No deployment is needed, but a Java installation must be present on the PC. They are effectively cached on the user computers, and they can provide sophisticated functionality. On the downside, each Java applet requires a Java virtual machine (JVM) to run. On terminal servers that serve multiple users at once, this requirement might lead to larger memory consumption.
- ▶ Plug-in viewers are native applications that must be installed through a setup routine on the user's computer. They integrate with the browser and provide their own viewing logic, which can be sophisticated (for example, with the AFP plug-in).
- ▶ The generic and Ajax viewers that are provided by Content Navigator provide limited rendering and viewing capabilities. They do not require any rollout or JVM.
- ▶ Transforms, such as the Ricoh AFP2PDF or other vendor-provided transforms, result in a PDF document that is viewed in the Acrobat viewer. Although this viewer is deployed on most user PCs, the rendering consumes processing power on the mid-tier system. Also, large documents cannot be rendered into PDFs. Because the PDF is displayed by an external application, it cannot communicate with the Content Manager OnDemand server like the line data applet.

Depending on the data that you are working with, consider these options:

- ▶ For Line Data:
 - The line data applet supports annotations. It can work with large object (LOB) reports if the large object functionality is employed at load time.
 - The Ajax viewer and direct rendering capabilities of Content Navigator work only on shorter reports. Additionally, the viewing of annotations and large object documents is not supported.
- ▶ For AFP data:
 - The AFP plug-in is the best choice, because it is almost identical to the client. However, it does not support annotations.

The only viewers that use this functionality are the line data applet, the AFP plug-in viewer, and the Content Manager OnDemand Windows client.

- AFP to PDF is a choice that does not require a plug-in rollout at the users' computers if the Acrobat plug-in is installed on their workstations. Font mappings must be configured at a central location. The additional workload on a rendering system and additional license costs must be considered. Large reports might not be able to be rendered or viewed.

Note: The AFP viewer plug-in, which is available with ODWEK and Content Manager OnDemand, is a version of the AFP viewer plug-in from the InfoPrint Solutions Company. Although the standard InfoPrint viewer can be used for viewing AFP, the ODWEK version uses direct communication with the Content Manager OnDemand server, enabling segmented document transfer for LOB documents.

Annotations

Only the native ODWEK viewers and the Windows client support annotations. These viewers and Windows clients support annotations in the following ways:

- ▶ Line data applet: Supports text. Starting with version 9, the viewer can work with graphical annotations, also.
- ▶ Windows Client: Supports maximum capabilities for all data types.
- ▶ Other viewers, for example, the AFP plug-in viewer: Do not support and are not aware of annotations.

Web clients, such as Content Navigator or the ODWEK Java API, can work with annotations and provide access to them through the hit list. Graphical annotations cannot be accessed that way because they are not exposed through the Java API.

Large object support

Large object (LOB) support is the methodology for working with large reports. For more information about how LOB affects your reports, see “Large object” on page 52.

From a viewer's perspective, if a large document is transferred, it generates high network traffic, resource consumption, and long wait times for users. If the viewer supports LOB documents, the viewer communicates with the server to transfer only the chunk of data that the user is looking at (for example, a 200 page chunk out of a 10,000 page report). If the user scrolls to a different chunk of pages, the viewer downloads only that relevant portion of the document that the user scrolled to.

The ODWEK Java API provides line-of-business operations. For more information, see *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646.

8.1.2 Client infrastructure options

Several basic architectural options, Windows client, Content Navigator, or API-based client integration into your line-of-business application, are available.

Windows client

Consider the following items when you are planning a Windows client infrastructure:

- ▶ It is faster than the web clients and more powerful.
- ▶ It requires native installation on each user's workstation or notebook. Server version upgrades might also require a new client installation.
- ▶ This client supports Citrix and Terminal services environments.
- ▶ It does not support the Transforms interface for transforming and converting data formats because the data formats are provided by ODWEK only.

Content Navigator

When you choose a ready-for-use web client, consider the IBM strategic client, IBM Content Navigator, because it is the most complete, most recent web client.

Special use cases might require the development of a custom client application for Content Manager OnDemand. For more information about development APIs, see 8.3, "Client API overview" on page 202.

With Content Navigator, you can run a cross-repository search to search for content across multiple types of repositories, including Content Manager OnDemand. For example, Content Manager OnDemand search results can be included in the same hit list as search results from other supported repositories to help provide a comprehensive view of content.

When you create a cross-repository search, you can specify the following information:

- ▶ Specify the scope of the search on each repository. You can specify the search or the classes that you want to include in the cross-repository search by using IBM Content Manager OnDemand. On IBM FileNet Content Manager and IBM Content Manager, you also can limit the search to a specific folder.
- ▶ Specify how properties from each repository are related to each other.
- ▶ Specify any default search criteria that you want displayed when users open the search.

For more information about how to configure a cross-repository search, see the IBM Content Navigator Knowledge Center at the following web address:

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3/contentnavigator_2.0.3.htm

Consider the following items when you choose Content Navigator or other clients:

- ▶ The viewers that are provided by Content Navigator are limited compared to the Windows client.
- ▶ A Content Manager OnDemand client that focuses only on Content Manager OnDemand is probably the easiest to maintain.
- ▶ A general Enterprise Content Manager Client, such as Content Navigator, with setup specifications that support the Content Manager OnDemand model and capabilities, might increase the dependency footprint of the client tier while it provides access to other systems through the same user interface.

Developing your own client

When you develop your own applications (web client), you can use the ODWEK Java APIs. For more information about the ODWEK APIs, see 8.3.1, “Content Manager OnDemand Web Enablement Kit” on page 202 and *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646.

If you are developing a Windows application, you optionally can use the Object Linking and Embedding (OLE) (ActiveX Control) API, which is provided by the Windows client. This API requires a Windows client installation.

Another option is to use an intermediate API that is based on the ODWEK Java API for the Content Manager OnDemand access portion. Content Management Interoperability Services (CMIS) or other web services can be used as the intermediate API. The web service application uses ODWEK to access Content Manager OnDemand and relays this access through its own web services to any other application. In this case, the Windows application only needs to talk to the web service. For more information about CMIS and its limitations, see 8.3.2, “Content Management Interoperability Services” on page 204.

The use of an intermediate API increases complexity and potentially decreases performance, but it decouples a Windows application and Content Manager OnDemand in terms of API versioning and requiring a Content Manager OnDemand installation.

8.1.3 Client compatibility

During the development history of Content Manager OnDemand, features were added and internal API schemes were changed. Therefore, not every client level can work with every server level. When you choose a client infrastructure for your Content Manager OnDemand environment, you must consider version dependencies.

Client compatibility matrix

At the API level, all user clients share a common API core that is based on the Windows client and ODWEK. Almost all other client and API implementations are based on these common APIs. An up-to-date overview of the compatibility matrix that shows the client and ODWEK level that can work with each server level is available at the following website:

<http://www.ibm.com/support/docview.wss?uid=swg21392275>

Determining version levels

Especially on IBM i and IBM z Systems, the server release level might not be obvious, because it is set by program temporary fixes (PTFs). The most convenient way to determine the server level of your Content Manager OnDemand system is to log on to either the Administrator Client or the user Windows client. After you are logged on, the clients show the server version in the status bar, as shown in Figure 8-1 on page 192.

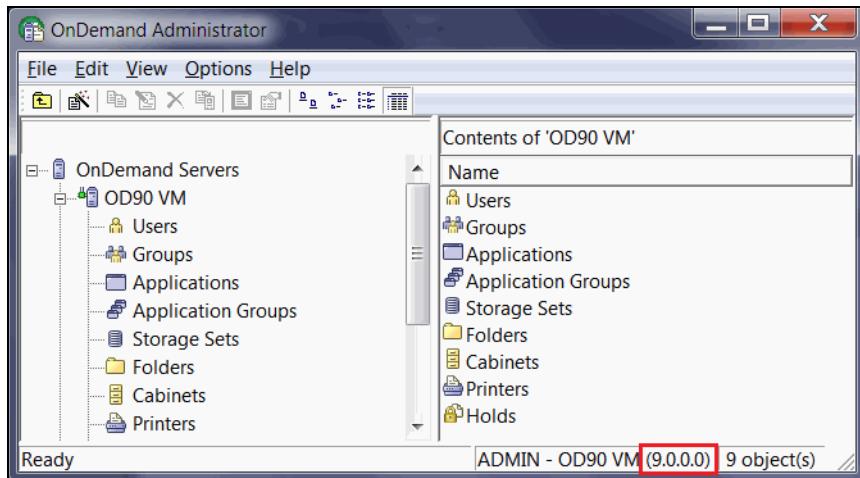


Figure 8-1 Server version is displayed in the client

Every **ars** command on the server displays its current server software version, as well.

You can view the version of the Windows client by clicking **Help → About**.

To determine the version of ODWEK, you can either look for the readme file in the ODWEK application directory or use a client. If you are running a web client (for example, Content Navigator), open a line data report by using the line data applet viewer. Because this viewer is provided by ODWEK directly, the viewer shows the current ODWEK version level in the About dialog box under the Help menu.

Cross-server calls with server console commands

Several of the **ars** commands that are provided by the server software installation, for example, the **ARSDOC** and **ARSLOAD** commands, can work with remote servers. This capability applies to cross-platform calls, for example, loading data with the **ARSLOAD** command that is running on Linux to a Content Manager OnDemand server that is running on the mainframe.

For more information, see “Server commands” on page 205.

Multiple versions at the same time

Before version 9.5, only one installation of the Content Manager OnDemand Windows client (user and administrative) was installed on a workstation concurrently. Multiple different versions were not allowed to coexist.

Starting with version 9.5 and later, you can run multiple versions of the Content Manager OnDemand Windows client (at the release level only, not the PTF level) on a single workstation. The client code is now installed in the `c:\Program Files (x86)\IBM\OnDemand Clients\V9.5` directory.

For ODWEK, you can run multiple versions of ODWEK on a single system. Although this capability might not be a preferred scenario from a maintenance point of view, it can be helpful during upgrades and existing system access scenarios. Each application that uses the ODWEK API must point to the correct installation path and load the correct corresponding libraries.

For more information, see the technote at the following website:

<http://www.ibm.com/support/docview.wss?uid=swg27019696>

8.2 Content Manager OnDemand Client options

In this section, we describe the common client options for Content Manager OnDemand, including web and non-web clients.

8.2.1 IBM Content Navigator

Content Navigator is the strategic client for IBM Content Manager, IBM FileNet P8, and Content Manager OnDemand. Access to Content Manager OnDemand servers is through the ODWEK Java API. Content Navigator is a Web 2.0 web client and requires a web application server, such as IBM WebSphere® Application Server.

Content Navigator can be used to access documents from multiple content repositories:

- ▶ IBM Content Manager Enterprise Edition repositories
- ▶ IBM Content Manager OnDemand repositories
- ▶ IBM FileNet P8 repositories
- ▶ Organization for the Advancement of Structured Information Standards (OASIS) CMIS repositories

With Content Navigator, users can perform these tasks:

- ▶ Search documents from any of the content repositories
- ▶ View documents side-by-side
- ▶ Edit document properties
- ▶ Add annotations to documents
- ▶ Send documents and document links through email
- ▶ Print documents
- ▶ Download documents

You can use Content Navigator to build a customized user experience. It supports many configuration options and includes a powerful API toolkit that you can use to extend the web client and build custom applications.

Figure 8-2 shows Content Navigator browsing a folder in Content Manager OnDemand.

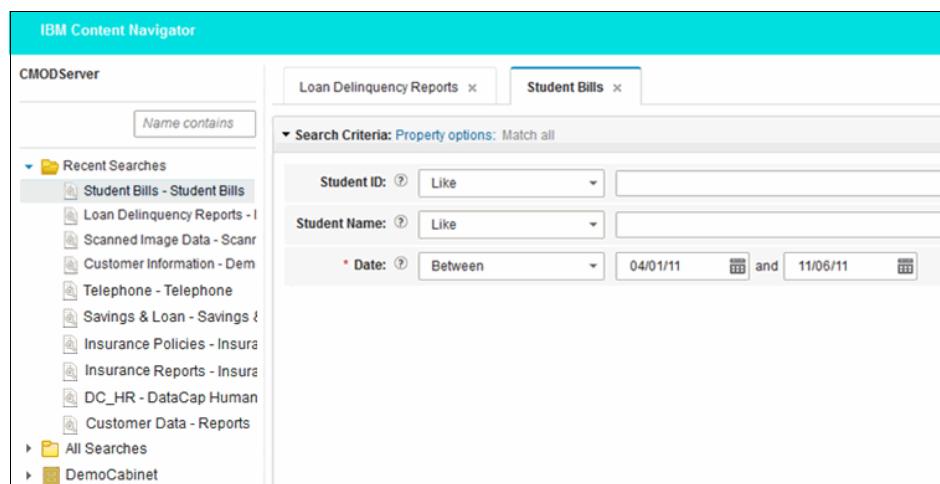


Figure 8-2 Searching a Content Manager OnDemand folder with Content Navigator

Content Navigator is a full-feature client for Content Manager OnDemand. Its interface follows modern user interface styles, with a browser pane on the left that shows the available Content Manager OnDemand folders, and a search and result pane on the right. All components and data are dynamic, and they can be resized and changed.

Note: Content Navigator is a Web 2.0 Ajax-based client. These web applications rely on an up-to-date JavaScript engine, which is only available in newer browsers. Older browsers, such as Microsoft Internet Explorer Version 8, might not work correctly with Content Navigator.

Content Navigator, version 2.0.2 and later, provides many additional Content Manager OnDemand capabilities:

- ▶ AFP Viewer plug-in support
- ▶ External Data Services (EDS) support
- ▶ Favorites support for folders and documents
- ▶ Single and multiple AFP file download as PDF (with AFP2PDF enabled)
- ▶ Highlighted search result terms in full text searches
- ▶ Line2PDF conversion viewer
- ▶ XML viewer

Starting with Content Manager OnDemand V9.0 Content Navigator provides single sign-on (SSO) token pass-through to the client side. Date validation is no longer required. Support is provided for ‘t’ date expression and federated search across Content Manager OnDemand, FileNet P8, and IBM Content Manager repositories. Content Navigator is also the new CMIS packaging for Content Manager OnDemand.

Installing Content Navigator

Content Navigator must be installed natively with ODWEK and IBM WebSphere Application Server (or any other applicable web application server). Typically, Content Navigator is installed on a separate system in the web tier and not on the same system as the Content Manager OnDemand server.

The following prerequisites exist for a Content Navigator installation for Content Manager OnDemand:

- ▶ Native installation of the Content Navigator base software
- ▶ A database to store the Content Navigator configuration
- ▶ Web application server
- ▶ ODWEK
- ▶ Optional: AFP Transforms for AFP to PDF rendering
- ▶ Java Database Connectivity (JDBC) drivers (if not already present)

The Content Navigator database is relatively small, so a collocation with the Content Manager OnDemand database might be possible in small deployments. The installation manual provides SQL statements for creating the database and its table spaces.

After you install all of the components, run the Content Navigator Configuration and Deployment Tool to create a preconfigured web application and deploy it to the web application server.

The Configuration and Deployment Tool provides a wizard that leads you through the base setup process. You must provide details about your web application server and connection information to the configuration database. For the Content Manager OnDemand configuration, you must provide the location of your ODWEK installation. Run the deployment scripts at the end for deploying Content Navigator on your application server.

The installation is described in detail in the “Planning, installing, and configuring IBM Content Navigator” section of the IBM Content Manager OnDemand Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUX_2.0.3/com.ibm.installingeuc.doc/eucao000.htm

Accessing the native libraries

The ODWEK Java API uses native libraries. To run Content Navigator (or any other web client that is based on the Java APIs), ensure that the web application server can access these libraries. To achieve this task, add the ODWEK directory into the **PATH** environment variable. On Windows platforms, you also must add the lib64 subdirectory of ODWEK into the PATH.

The following example shows the path to the directory in Windows:

```
PATH=%PATH%;C:\Program Files\IBM\OnDemand\V9.5\www;C:\Program Files\IBM\OnDemand\V9.5\bin
```

On Linux and UNIX platforms, it is necessary to expand the **LD_LIBRARY_PATH** (LIBPATH on AIX) to include the ODWEK directory. This step must be performed in the environment on which the web application server is running by editing the start scripts.

For example, on Linux, you run this command:

```
export LD_LIBRARY_PATH="/opt/ibm/ondemand/V9.5/www:$LD_LIBRARY_PATH
```

The Content Navigator installer creates a shared native library in WebSphere Application Server. You can review this library in the Integrated Solution Console in the Environment, Shared libraries section. You need a library that has the class path set to the location of the **ODApi.jar** (for example, /opt/ibm/ondemand/V9.5/www/api/ODApi.jar) and the Native Library Path set to the ODWEK directory (for example, /opt/ibm/ondemand/V9.5/www). If you encounter any errors, ensure that these paths are valid.

Note: If multiple applications reference the same native library, the library gets loaded multiple times. But because the ODWEK library is a shared library, it can be loaded only one time for each JVM. So, if you are running multiple ODWEK web applications in one WebSphere Application Server, you must configure the shared library reference on the Class Loader level of the server itself instead of on the application level. You can use the Integrated Solution Console, which is in the class loader of the application server, for this task.

Administering Content Navigator

Content Navigator administration is performed in the admin desktop of the Content Navigator web application. For more information, see the “Administering IBM Content Navigator” section of the IBM Content Manager OnDemand Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUX_2.0.3/com.ibm.installingeuc.doc/eucco037.htm

Adding a Content Manager OnDemand repository to Content Navigator

Multiple Content Manager OnDemand repositories can be added to a Content Navigator installation, exposing each repository to a defined set of users through the configuration of different desktops.

For the configuration of Content Manager OnDemand repositories, you need the following parameters:

- ▶ Display name: The depository name that is displayed to the users.
- ▶ Server name: IP or host name of the Content Manager OnDemand server.
- ▶ Port number: Instance port (the default is 1445).
- ▶ If you want an encrypted connection between ODWEK and the Content Manager OnDemand server, enable Secure Sockets Layer (SSL) and provide an SSL key ring database and stash file. Enabling SSL consumes *additional resources* on both systems (Content Manager OnDemand and the web tier).

Note: This option does not affect the SSL security of the web application, for example, between the web server and the browser. It only encrypts the API communication between the web tier and the Content Manager OnDemand server.

- ▶ If you want to use AFP Transforms or another transform filter through generic transforms, you must specify the path to the correct configuration files.

You can specify additional configuration parameters, for example, in the ODConfig class in the Java API. For more information, see the Javadoc of ODApi or *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646.

Content Navigator viewer options

For each Content Navigator Desktop, a different viewer map can be active. Within a viewer map, for each content type, a different viewer can be configured. Several viewers are available to Content Manager OnDemand repositories in Content Navigator:

- ▶ Content Navigator uses the viewers that ship with ODWEK, for example, the line data applet. Repository-specific features can be handled only by ODWEK viewers.
- ▶ ODWEK performs conversions, for example, an AFP to PDF conversion.
- ▶ Built-in viewers for Content Navigator:
 - Ajax viewer and a simple PDF and HTML conversion
 - Web browser pass-through
 - PDF-inline viewer for addressing the Adobe Acrobat viewer browser plug-in
 - Generic Applet viewer
 - IBM Daeja™ ViewONE Virtual viewer

For a full listing of the viewers, see the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEUEX_2.0.3/com.ibm.installingeuc/eucco002.htm?lang=en

- ▶ Content Manager OnDemand plug-in viewers for Content Navigator:
 - AFP Viewer plug-in
 - FileNet Content Federation Services Viewer plug-in
 - XML Viewer plug-in
- ▶ Third-party viewers can be integrated into Content Navigator. For IBM Production Imaging Edition, for example, a third-party viewer is integrated with Content Navigator. You can integrate your own viewer by using the Content Navigator plug-in architecture.

The generic applet viewer (“applet viewer”) is a Java applet, which can handle various types of documents, such as PDF and Microsoft Office documents (which it renders), images, line data, and AFP documents. The generic applet viewer might be an option if you work with images that are stored in Content Manager OnDemand.

If you want to avoid the use of Java applets and your content is viewable by browsers (for example, certain image types or textual data), try the browser pass-through viewer, which lets the browser handle the data natively. If you work with AFP and must use the AFP browser plug-in, register the Content Navigator plug-in, `AFPViewerPlugin.jar`, and configure the viewer map that is assigned to your Content Navigator desktop to use the AFP viewer for the application/afp MIME type. The `AFPViewerPlugin.jar` file ships with Content Navigator. You must choose the web browser pass-through viewer.

The Ajax viewer is a Web 2.0 JavaScript application that provides basic document functions, such as page-wise browsing, rotation, or zoom. It is not a Java applet.

The generic applet viewer, the built-in PDF and HTML conversion, and the Ajax viewer can all work with various data types:

- ▶ Images (such as TIFF, JPEG, and DICOM)
- ▶ Office documents
- ▶ PDF
- ▶ Most line data documents
- ▶ Certain AFP data

However, they all use a rendering engine to display Office, PDF, and AFP data into an image. This rendering might work well with certain Office and PDF files, but it fails on most non-basic AFP data streams.

For more information, see 8.1.1, “Viewer options” on page 186.

Note: Content Navigator is a Web 2.0 client and relies on HTML 5 and JavaScript for its core client functionality and especially for the Ajax viewers. Not all browsers are suitable for running Content Navigator fast and efficiently, especially for Microsoft Internet Explorer browsers before version 9. Test Content Navigator with your user browser thoroughly before you consider a deployment.

Extending Content Navigator

Content Navigator is not designed as a client that is dedicated solely to Content Manager OnDemand, so a more complex configuration is necessary than with simpler client options. Content Navigator provides many configuration and customization options through its API and plug-in methodology. For more information about the customization options of Content Navigator, see *Customizing and Extending IBM Content Navigator*, SG24-8055.

8.2.2 Content Manager OnDemand Windows client

The Content Manager OnDemand Windows client is a full function, feature-rich client that meets the needs of line-of-business application areas and customer service representatives. The Windows client displays content in its native format and is considered a corporate internal access client. Many technical aspects of the Windows client are described in 8.1.1, “Viewer options” on page 186 and 8.1.2, “Client infrastructure options” on page 190.

Figure 8-3 shows a user that is logged in to a folder that performed a search and received the results list. Figure 8-3 shows the indication of a note or hold and also the location of the document. On the right side of the hit list, the load date and document size are displayed.

The screenshot displays the Content Manager OnDemand interface. The top section, titled "Search Criteria", contains six input fields with dropdown menus for operators (Like, Equal To, Between) and text input fields for "Form Number", "Book Name", "Text Search", "Page Count", "Doc Size", and "Load Date". The bottom section, titled "Document List", shows a table of search results. The columns are labeled "Form Number", "Book Name", "Page Count", "Doc Size", and "Load Date". The first two columns have circled icons above them: a yellow square with a blue border and a blue square with a white border. The "Doc Size" column has a circled "M" icon above it, indicating megabytes. The table data includes:

Form Number	Book Name	Page Count	Doc Size	Load Date
S544-5293-01	Administrator's Reference	165	991118	2012-01-01
S544-5463-00	Getting Started with the Administrator's Reference	174	730398	2012-01-01
S544-5469-00	Getting Started with the User Interface	31	278190	2012-01-01
S544-5489-00	Indexing Reference	196	1304602	2012-01-01
G544-5280-01	Installation and Configuration Guide	250	1016962	2012-01-01
G544-5281-01	Introduction and Planning Guide	134	2286996	2012-01-01
S544-5278-01	OS/2 Client User's Guide	145	8474131	2012-01-01
S544-5275-01	Windows Client User's Guide	129	642650	2012-01-01

Figure 8-3 Content Manager OnDemand results list in the Windows client

As the full function client for Content Manager OnDemand, the Windows client provides various business functions and features that can be selected at the document level, as shown in Figure 8-4 on page 199.

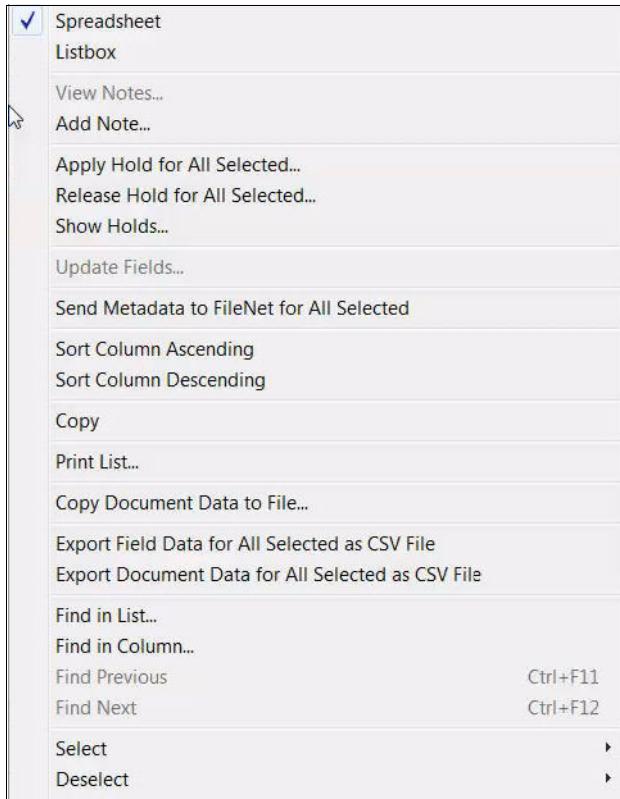


Figure 8-4 Windows client capabilities

You also can show the pages within a document or report as *thumbnails*, which provide you with a visual representation of the report.

8.2.3 CICS Client

The CICS 3270-based interface was the original user interface for Content Manager OnDemand z/OS. It was the predecessor to the Windows and web technology clients that are used today by most Content Manager OnDemand customers. Customers still request the CICS Client to use it to meet their production needs, typically as they migrate their user base (and applications) from a host environment to a client/server or web architecture. The CICS client was developed to meet this need. The CICS Client provides a functional subset of the windows and web clients. The CICS Client is English only. It is included in the Content Manager OnDemand maintenance. It does not ship in the Content Manager OnDemand package, so it must be downloaded and installed separately.

The CICS Client can be downloaded from the following website:

<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=db2cmto>

Figure 8-5 on page 200 shows the Content Manager OnDemand CICS Client login panel, which requires the standard login credentials.

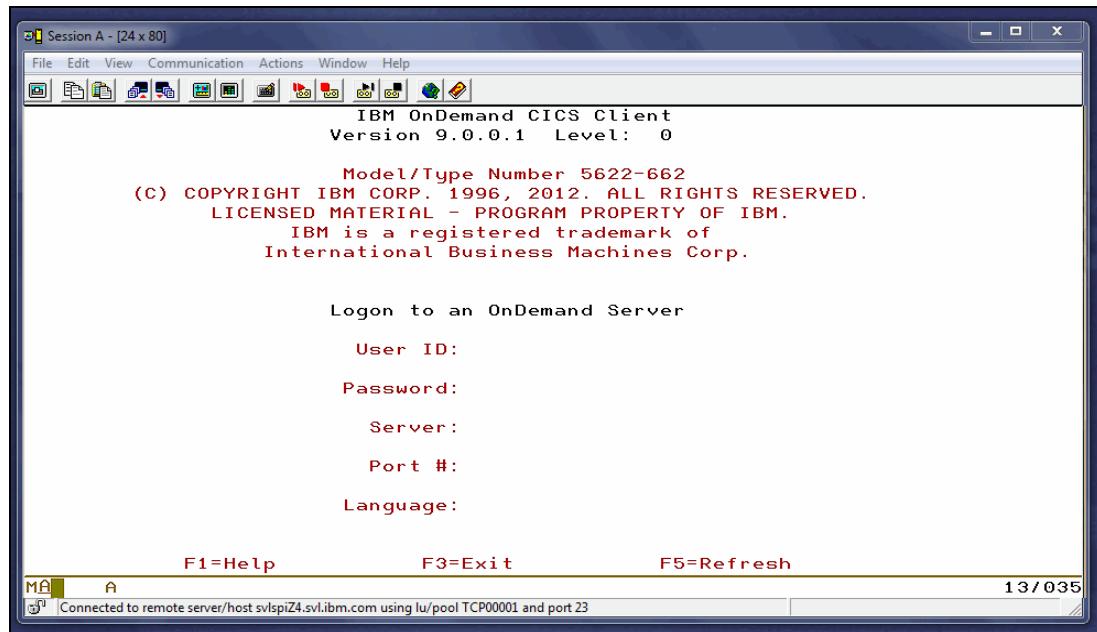


Figure 8-5 Content Manager OnDemand CICS Client login panel

The CICS Client provides viewing capabilities for line data reports and a “best fit” model for fully composed AFP documents. Viewing a standard line data report is shown in Figure 8-6.

REPORT D94100100		PENNANT NATIONAL BANK		DATE	
BANK	001	FROM	10/01/94	TO	10/01/94
LOAN DELINQUENCY REPORT					
LOAN NUMBER	CUSTOMER NAME	LOAN AMOUNT	DELINQUENT 30 DAYS	DELINQUENT 60 DAYS	DE
0000010000	MCMULLIGAN, PATRICK	\$10000000.00	\$ 50.00	\$ 50.00	\$
0000010001	ABBOTT, DAVID	\$ 11000.00	\$ 100.00	\$ 200.00	\$
0000010002	ABBOTT, DAVID	\$ 12000.00	\$ 140.00	\$.00	\$
0000010003	ABBOTT, DAVID	\$ 13000.00	\$ 150.00	\$.00	\$
0000010005	ROBINS, STEVEN	\$ 500.00	\$ 50.00	\$.00	\$
0000010006	PALMER, ARNOLD	\$ 1000.00	\$ 75.00	\$ 150.00	\$
0000010007	PETERS, PAUL	\$ 650.00	\$ 50.00	\$.00	\$
0000010008	ROBERTS, ABRAHAM	\$ 9000.00	\$ 120.00	\$.00	\$
0000010009	SMITH, RANDOLPH	\$ 8000.00	\$ 115.00	\$.00	\$

Figure 8-6 Viewing a standard line data report

8.2.4 Integration with other Enterprise Content Manager products

Content Manager OnDemand provides integration points with other IBM Enterprise Content Manager software on many different levels. Integration can occur on the client level (for example, by using another product's user interface (UI) as the client for Content Manager OnDemand). You can use an infrastructural integration in which another product accesses Content Manager OnDemand and information is exchanged between the products at a lower level.

For more information about the most common integrations, see *Federated Content Management: Accessing Content from Disparate Repositories with IBM Content Federation Services and IBM Content Integrator*, SG24-7742.

8.2.5 Federated search with IBM Information Integrator

Information Integrator is an IBM Enterprise Content Manager product that is available for all Enterprise Content Manager customers. Although it has many functions, it is primarily a federation system.

It can connect to various systems, such as Content Manager OnDemand, Content Manager, FileNet P8, and content management systems by other vendors. You can create a virtual archive, spanning across all connected systems and document models. Users can search in one system and the search is propagated to multiple back-end repositories. Information Integrator maps virtual fields to folder fields in Content Manager OnDemand (or respective models in other systems) and delivers a consistent hit list of documents to the user.

Content Integrator might be an option for you if you use separate Content Manager OnDemand systems (instances or physical systems) and must provide a cross-system search (for example, for eDiscovery or legal inquiries). Another use case is to provide repository-neutral services with access to multiple content management systems.

Note: Information Integrator is an abstraction layer. You lose Content Manager OnDemand specific functionality, because the virtual archive provides only the common functionality that can be implemented by all archives. Always check your use case to verify that a virtual archive meets your needs for functional compatibility and performance.

8.2.6 Integration with IBM FileNet P8

Integration exists between IBM FileNet P8 and Content Manager OnDemand through FileNet Content Federation Services. Content Manager OnDemand documents can be federated into FileNet P8, making them accessible like any other FileNet P8 documents for FileNet P8 users.

This federation differs compared to Information Integrator. In Content Federation Services, for each Content Manager OnDemand document, a virtual document is created in FileNet P8 (resulting in database records in FileNet P8). So, these documents act as FileNet P8 documents from a FileNet P8 user's perspective. Information Integrator does not have its own database and does not create virtual documents, but it instead calls Content Manager OnDemand for searches and passes on the result list. A search in FileNet P8 never starts a search in Content Manager OnDemand, but it can find only federated Content Manager OnDemand documents, which are cataloged in the FileNet P8 database.

If a FileNet P8 system is installed in your environment that serves as your primary content management system and reports need to be available to users without their knowing that those reports are in a different system, this integration might suit your needs. The same situation applies to the use of FileNet P8 Records Management, which can be applied to Content Manager OnDemand documents as well, therefore bringing a level of federated records management capability to your documents.

When you plan your integration with FileNet P8, remember this federation is active: Content Manager OnDemand actively publishes document links into a FileNet P8 system. You must consider both volumes (FileNet P8 systems usually are smaller than Content Manager OnDemand systems) and the active federation process.

For more information about Content Manager OnDemand and FileNet P8 integration, see *IBM FileNet Content Federation Services for Content Manager OnDemand*, SC19-2711.

8.3 Client API overview

With various client options, multiple API options are available to navigate through the system and access Content Manager OnDemand documents. Although the Java API that is provided by Content Manager ODWEK is the API that is used most by clients and the basis for most development projects, other APIs are available and used for a limited range of scenarios.

The following list shows the APIs that are available for Content Manager OnDemand:

- ▶ Content Manager ODWEK: The Java API for Content Manager OnDemand
- ▶ SOAP and Representational State Transfer (REST) web services that follow the CMIS standard
- ▶ Windows OLE (ActiveX control) that is provided by the Windows client
- ▶ XML administrative API through the **ARSXML** server command
- ▶ Structured APIs on z/OS environments
- ▶ The standard Content Manager OnDemand server commands that serve as a console-based API to work with Content Manager OnDemand documents

8.3.1 Content Manager OnDemand Web Enablement Kit

ODWEK provides a Java API to access Content Manager OnDemand servers and their documents. It is the strategic client API that provides the largest feature set of any Content Manager OnDemand API. It is used by web clients, such as Content Navigator or WEBi, by abstraction layers, such as Information Integrator, or by API components, such as CMIS.

The ODWEK Java API and its use to develop Content Manager OnDemand clients are described in detail in *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646. This section covers only a basic overview and focuses on client considerations about ODWEK. Developers are encouraged to read the referenced book before they plan a client development that is based on ODWEK.

Scope

ODWEK is a Content Manager OnDemand component that can be used by all Content Manager OnDemand customers. It is focused on typical client use cases, such as searching for and accessing data that is stored in a Content Manager OnDemand archive. It also has web viewers, such as the line data applet and Content Manager OnDemand AFP viewer.

For more information about ODWEK viewers and conversion support, see “Windows client viewers” on page 187.

Before Content Manager OnDemand Web Enablement Kit (ODWEK) Java API V9.5, the only API that allowed documents to be added to the Content Manager OnDemand archive was the `ODFolder.storeDocument` API, which resulted in an archive request to the Content Manager OnDemand server for each document. This API is suitable for low-volume ad hoc storage.

In ODWEK V9.5, new APIs were introduced to allow documents to be loaded in bulk, which provides high-volume storage similar to the `arsload` command. To accomplish bulk loading by using the ODWEK Java API, you perform these steps:

1. Call the `ODServer.loadInit` API to initiate the load process.
2. For each document to load, call the `ODServer.loadAddDoc` API, which passes the number of pages, a hash table of index values to store, and the document data.
3. Call the `ODServer.loadCommit` API, which specifies the application group and application to send the load data and load request to the Content Manager OnDemand server.

For special client needs, the Java API provides access to the object model (application group and application) of Content Manager OnDemand and facilitates an `ARSXML` pass-through, which can be used to perform administrative tasks.

Native library dependency

Because of the nature of the Content Manager OnDemand architecture, ODWEK requires the use of native libraries.

In addition to the physical presence on the system, Java applications must be aware of the native libraries. The ODWEK native libraries are loaded as shared memory objects and cannot be reloaded multiple times. If you run multiple ODWEK applications in one web application server, consider this restriction.

For a description of how the native library reference is managed for the ODWEK client in IBM Content Navigator in IBM WebSphere Application Server, see “Accessing the native libraries” on page 195.

ODWEK web client design considerations

When you design a web client for Content Manager OnDemand that is based on ODWEK, consider the following items:

- ▶ Dependency on a native shared library affects deployment and general options, such as the message language, which can be set only for the whole environment.
- ▶ Be careful with multithreading document access. Access to a single session with the Content Manager OnDemand server must be in a single-threaded fashion. Only one thread can access objects of a specified Content Manager OnDemand session at a time.
- ▶ Every session that is established with a Content Manager OnDemand server consumes memory on the ODWEK system. For high-usage applications that support many concurrent users, for example, web clients that work with non-named users, we suggest the use of connection pooling.
- ▶ Ensure that a timeout concept is implemented in your application that meets the Content Manager OnDemand user activity timeout. Sessions that do not time out might lead to memory leaks or high memory consumption on the Content Manager OnDemand and ODWEK machines.

Note: Starting with version 9 of ODWEK, additional functions were added to reset the inactivity timeout counter of a user session. This enhancement simplifies the design of connection pooling and timeout scenarios.

For a connection pooling sample that covers the topics of thread safety, resource consumption, and timeouts in detail, see Chapter 6, “Connection pooling and connection handling”, in *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646.

8.3.2 Content Management Interoperability Services

CMIS is an open standard for accessing content management repositories. It is an OASIS specification and it is supported by various applications from different vendors, including IBM (with FileNet P8, Content Manager, and Content Manager OnDemand).

CMIS provides a common access interface for searching, retrieving, and in the case of document management systems, modifying and deleting documents. It is a web services interface that is implemented in either SOAP web services and REST (Atom) services.

For more information about CMIS, see the CMIS page on the OASIS website, the CMIS overview page at the IBM Enterprise Content Manager website, and the technical documentation that is available:

- ▶ <https://www.oasis-open.org/committees/cmis/>
- ▶ <http://www.ibm.com/software/ecm/cmis.html>
- ▶ *Implementing Web Applications with CM Information Integrator for Content and OnDemand Web Enablement Kit*, SG24-6338
- ▶ Content Management Interoperability Services for Content Manager OnDemand is installed as part of the IBM Content Navigator installation. For more information, see “Installing Content Navigator” on page 194.

When you consider implementing your own software on CMIS, remember CMIS is used for accessing document management systems, but not necessarily high-volume report archives, such as Content Manager OnDemand.

The methodology of accessing documents is based on folders and subfolders with documents in it (such as in a file system) and partially emulated by Content Manager OnDemand with its different object model. The use of CMIS must be considered as an abstraction layer that might have an impact on throughput and feature exposure. Also, much of the CMIS API is not supported by Content Manager OnDemand (such as the storage and deletion functions).

8.3.3 Other client-based API options

Other client-based API options include Windows ActiveX API, structured API on z/OS, server commands, and XML Administration interface (ARSXML).

Windows ActiveX API

The Windows client ships an ActiveX control, which can be used in its own application for accessing Content Manager OnDemand servers and documents through the functions that are provided by the Windows client. It is a development API that enables the development of custom applications that use an installed Windows client as the API provider. The ActiveX API covers only a basic operation subset.

For more information about the Windows client-based API, see *Windows Client Customization Guide*, SC19-3357.

Structured API on z/OS

In z/OS environments, Content Manager OnDemand includes Structured APIs that provide custom applications in CICS, IBM IMS™, TSO, or batch environments with the ability to connect to Content Manager OnDemand servers. The Structured APIs support only the basic read operations (log on, open folder, search, and retrieve documents and annotations).

Structured APIs are handled by a dedicated component of Content Manager OnDemand that is called *MidServer*. MidServer relies on ODWEK and its API to access the Content Manager OnDemand server.

Structured APIs are available only on z/OS, and they are called from COBOL or C applications in the same manner as MVS calls. Because ODWEK is used as the access path to the Content Manager OnDemand server, the Structured APIs can be used to access non-z/OS Content Manager OnDemand servers, as well.

Server commands

In addition to the API options, which are exposed through Java, OLE, or Web Services, Content Manager OnDemand provides console (command-line) applications that provide specific functions, such as searching, retrieving, or deleting documents, and sophisticated functions, such as placing holds and working with the full text engine. Most of this functionality is exposed through the **ARSDOC** application.

Simpler custom applications, for example, shell scripts, can use these server console applications to interact with Content Manager OnDemand systems. The applications are available only as part of a Content Manager OnDemand server installation. Because most of them (namely **ARSDOC**) communicate with the server through TCP/IP, you can connect and interact with Content Manager OnDemand servers remotely on other platforms. When you call remote servers, ensure that the local installation that provides the ARS applications and the actual Content Manager OnDemand server are on the same version level.

For more information about the administrative commands, see the specific command descriptions in the IBM Content Manager OnDemand Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSEPCD_9.0.0/com.ibm.ondemandtoc.doc/administering.htm

XML administration interface: ARSXML

In addition to the user client APIs, the **ARSXML** server command provides an interface for administrative users and applications to access the Content Manager OnDemand data model. By using **ARSXML**, folders, application groups, applications, and users can be exported, created, deleted, and modified. It works on XML documents by describing the change, action, or selection criteria and the resulting output XML document.

ARSXML is a console application that is available on the Content Manager OnDemand server. It can work with remote servers if they are at the same release level.

XMLs can be passed to and from **ARSXML** through the ODWEK Java API, which enables Java applications to programmatically call **ARSXML** and obtain access to administrative data model functions.



Data conversion

In this chapter, we provide information about data conversion for IBM Content Manager OnDemand (Content Manager OnDemand). We describe the reasons for data conversion and describe the interface that Content Manager OnDemand uses to convert data.

In this chapter, we cover the following topics:

- ▶ Overview of data conversion
- ▶ Generic Transform Interface

9.1 Overview of data conversion

To work with data conversion, understand the data conversions that are required, and when and how to convert the data. Perform detailed planning before you build your solution so that you achieve a design that remains efficient for many years.

In this section, we describe why you might need data conversion, when to convert the data stream, and how to convert the data.

9.1.1 Why convert data streams

You might want to convert data streams for many reasons:

- ▶ Certain data streams, such as Hewlett-Packard (HP) Printer Command Language (PCL) or Xerox metacode, are printer-specific and cannot be displayed. Before you archive or display the documents, these data streams must be transformed into a compatible format.
- ▶ The archived data stream might need to comply with a company's internal rules or regulations. Therefore, the produced data streams must be transformed into the defined and required final format before they are archived.
- ▶ The documents might need to be accessible by a user that is outside of the company. The document must be displayed through standard tools that are available on any or at least most of the clients, such as an Internet browser or Adobe Acrobat Reader.
- ▶ The documents might need to be manipulated so that only part of the document is displayed in a personalized way.

9.1.2 When to convert data streams

The decision of *when* to convert data streams relies mainly on the use of the system. Typically, converting data at load time requires more time to process the print stream file, and converting data at retrieval time causes the user retrieval to be a little slower. The decision might depend on how many documents are retrieved, compared to how many documents are loaded daily. It might also depend on legal requirements about the format of stored data.

AFP to PDF

If a requirement exists to present AFP documents in the Portable Document Format (PDF) format over the web, from a storage perspective, it is more efficient to store the documents in their native format and then convert them to PDF at retrieval time. AFP documents are stored more efficiently than PDF documents.

The PDF print stream, when it is divided into separate customer statements, is larger than AFP because each statement contains its own set of structures that are required by the PDF architecture to define a document.

Elapsed time and processor time are also essential factors in the decision-making process. The amount of time (elapsed and CPU) that is needed to convert the document depends on how large the document is and how many resources or fonts are associated with the document.

9.1.3 How to convert the data

Content Manager OnDemand uses the Generic Transform Interface to integrate Content Manager OnDemand with third-party transform solutions.

Consider the following information about target flows:

- ▶ HTML might be used with the same intent, but an HTML document is not always displayed identically, depending on the web browser that is used. Additional testing that includes your needs and the encountered environments might be necessary for validation before the implementation.
- ▶ PDF might be used as a way to make documents available through standard and no-charge tools, such as Adobe Acrobat Reader. The transformed documents must be displayable, saveable, and printable the same way regardless of the environment on which the user works.
- ▶ XML is an intermediate text-based data format for the manipulation of documents, regardless of the source data stream, and displays the documents totally or partially in a personalized way. The use of XML usually involves additional development, including scripts and stylesheets.

9.2 Generic Transform Interface

Content Manager OnDemand uses the Generic Transform Interface to manage third-party data transforms for the Content Manager OnDemand Web Enablement Kit (ODWEK) application programming interface (API) set. This interface is used with the document retrieval APIs.

The ODWEK Java API provides industry-standard Java classes that can be used by a customer to write a custom web application that can access data that is stored on the Content Manager OnDemand server. This custom application can, for example, permit the user to log on to a Content Manager OnDemand server, get a list of folders, search a specific folder, generate a hit list of matching documents, and retrieve those documents for viewing. Many APIs provide advanced functionalities.

For more information, see the following resources:

- ▶ *IBM TechDoc Best practices for building Web Applications using IBM Content Manager OnDemand Java APIs:*

<https://www.ibm.com/support/techdocs/atstr.nsf/WebIndex/WP101203>

This document, which is prepared by the Content Manager OnDemand development team, provides recommendations about how to use the ODWEK Java APIs. Use this document to understand how the ODWEK Java APIs interface with the Java virtual machine (JVM) and Content Manager OnDemand systems to avoid common coding mistakes.

- ▶ *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646:

This publication provides basic and advanced information about how to use the ODWEK Java APIs to develop custom applications.

9.2.1 Overview

Before version 8.5.0.0, the ODWEK Java APIs provided a tight integration with only a few specific transforms: AFP2PDF, AFP2HTML, and AFP2XML. These transform engines were used by ODWEK clients to generate different document types for display purposes. Although this capability provided invaluable functionality, it meant that new transform engines were not readily integrated into ODWEK.

To meet this requirement, a highly flexible interface was added to the ODWEK Java APIs that allows a developer to easily implement a third-party document transform solution.

The new ODWEK Interface allows a client developer to implement an external program to transform a document in one of two ways:

- ▶ If the transform vendor provides a basic command-line executable file, it is implemented in an XML interface, which supports the retrieval of all of the document details that are stored in Content Manager OnDemand, and also allows specific options to be passed to the transform.
- ▶ The ODWEK Java APIs also provide a Java interface that a client developer can use to add even more flexibility to their client solution. The Java interface allows a client developer to get the document byte stream from ODWEK, then use any methods that they want to convert the document. These methods can include calls to web services that allow remote transformation. After the document is transformed, the resulting data can be returned to ODWEK, where it is passed back to the client that made the request.

9.2.2 Configuration

To enable the Generic Transform Interface in ODWEK, an XML document must be created and defined in the `ODConfig.Properties` object. This XML document is identified by the `<ODConfig.TransformXML>` key name and must include the fully qualified path to the XML file where the transforms are defined.

After you configure your XML configuration for the Generic Transform Interface, as described in 9.2.3, “Basic implementation: Executable interface” on page 211, you can enable this functionality in your ODWEK environment, as shown in Example 9-1.

Example 9-1 Enabling Generic Transform Interface in the ODWEK environment

```
Properties props = new Properties();
props.setProperty(ODConfig.TRANSFORMS_XML, "transform.xml"); /*Fully qualified
path to XML file containing transform details.*/
ODConfig odConfig = new ODConfig(ODConstant.PLUGIN, //AfpViewer
                                ODConstant.APPLET, //LineViewer
                                null, //MetaViewer
                                10, //MaxHits
                                "", //AppletDir
                                "ENU", //Language
                                null, //TempDir
                                "c:\tracedir", //TraceDir
                                4, //TraceLevel
                                props); //Additional properties
```

9.2.3 Basic implementation: Executable interface

The basic implementation of the Generic Transform Interface involves an XML configuration to define a transform to ODWEK that uses the command-line (`cmdline`) executable functionality. With this configuration, you can request details that Content Manager OnDemand stored for the document to be passed in the specified `cmdline` options and to also pass through transform-specific options, as specified in the `ODTransform.xml` file.

Example 9-2 shows a sample of the `ODTransform.xml` file that can be used in this implementation.

Example 9-2 ODTransform.xml sample

```
<Transforms>
  <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension><CmdParms>
      <RECORDLENGTH>-lm</RECORDLENGTH>
      <CARRIAGECONTROL>-x</CARRIAGECONTROL>
      <CODEPAGE>-a</CODEPAGE>
      <OUTPUTFILE>-o</OUTPUTFILE></CmdParms>
      <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
      <Passthru>
        <!-- Use tag cmdlineparm to declare additional cmdline variables that the
        transform might require -->
        <Cmdlineparm>-r PDF</Cmdlineparm>
      </Passthru>
    </transform>
<Transforms>
```

In this example, you can see that we defined a transform that is named `MyTXFRM_EXE`, which calls the transform command `txfrm.exe`, which is defined in the `<CmdLineExe>` tag.

The `<TransformName>` is used as the viewer name when it calls the ODWEK Retrieve APIs. From this configuration, ODWEK knows that the transform requires RECORDLENGTH, CARRIAGECONTROL, CODEPAGE, and OUTPUTFILE information from Content Manager OnDemand, and can set it on the `cmdline` by using the options that are specified in each related XML tag.

Also, the `txfrm.exe` requires additional information to be passed on the `cmdline`. The `-r` that is specified in the `<Cmdlineparm>` tag has no meaning to Content Manager OnDemand, so it is passed through and set on the `cmdline` call to the `txfrm.exe`.

In the custom Java code, the call to retrieve the data from ODWEK includes the `<Transform Name>` that is specified in the XML and looks like the following line:

```
"byte[] transformedDocument = ODHit.retrieve("MyTXFRM_EXE");
```

From this example definition, ODWEK calls the specified transform with the following `cmdline` executable file. Details for the items within “`< >`” are provided by ODWEK from the Content Manager OnDemand data definitions:

```
"c:/opt/txfrm.exe -lm<record len> -x <carriage control> -a <codepage> -o <output
file name> -r PDF"
```

9.2.4 V9.5 enhancement: Customizing values that are returned from ODWEK

For certain transforms, values that are returned from ODWEK might not be consistent with the command-line values that are expected by the transform. For example, a transform might have a fixed set of options to specify a carriage control type. The values that are returned by ODWEK when the <CARRIAGECONTROL> tag is included in the <CmdParms> are 'A' (ANSI), 'M' (Machine), and 'N' (None). The following command is produced by the XML in Example 9-1 on page 210:

```
c:/opt/txfrm.exe -lm 133 -x A -a 500 -o <outputfilename> -r PDF <datafilename>
```

Because the <CARRIAGECONTROL> tag is present, ODWEK returns the document's corresponding value - "-x A", or "-x M", or "-x N", depending on the carriage control type (CC Type) that is defined in this document's application definition. If the transform defines a different set of acceptable values, for example 2, 4, and 0, to specify the document's carriage control, you can map those values by substituting the following XML as shown in Figure 9-1.

```
<Transforms>
  <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <ODFileType>L</ODFileType>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdParms>
      <RECORDLENGTH>-lm</RECORDLENGTH>
      <CCANSI>-x 2</CCANSI>
      <CCMACHINE>-x 4</CCMACHINE>
      <CCNONE>-x 0</CCNONE>
      <CODEPAGE>-a</CODEPAGE>
      <OUTPUTFILE>-o</OUTPUTFILE>
    </CmdParms>
    <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
    <Passthru>
      <!-- Use tag cmdlineparm to declare additional cmdline variables that the tran
might require -->
      <Cmdlineparm>-r PDF</Cmdlineparm>
    </Passthru>
  </transform>
```

Figure 9-1 Sample XML with custom options

Note: The <CARRIAGECONTROL> node was replaced by three values. When the CC Type that is returned by ODWEK matches ANSI, rather than an 'A', the command includes "-x 2".

This type of substitution can be used to specify the RECFM (Record Format), PRMode, TRC, and CC Type.

9.2.5 V9.5 enhancement: Application Group and Application-specific XML

In version 9.5.0.2, ODWEK now provides additional options under the <transform> node that allow the transform command parameters to be generated based on an Application Group, or an Application Group and Application pair.

Figure 9-2 on page 213 shows a sample transform.xml that can be used in this implementation.

```

<Transforms>
  <transform>
    <TransformName>MyTXFRM_EXE</TransformName>
    <TransformDescription>Transform Cmdline Executable</TransformDescription>
    <ODFileType>L</ODFileType>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdLineExe>c://opt//txfrm.exe</CmdLineExe>
    <CmdParms>
      <RECORDLENGTH>-lm</RECORDLENGTH>
      <CARRIAGECONTROL>-x</CARRIAGECONTROL>
      <CODEPAGE>-a</CODEPAGE>
      <OUTPUTFILE>-o</OUTPUTFILE>
    </CmdParms>
    <Passthru>
      <!-- Use tag cmdlineparm to declare additional cmdline variables -->
      <!-- that the transform might require -->
      <Cmdlineparm>-r PDF</Cmdlineparm>
    </Passthru>
    <ApplicationGroup name='FinancialReports'>
      <CmdParms>
        <CARRIAGECONTROL>-x</CARRIAGECONTROL>
        <CODEPAGE>-c</CODEPAGE>
        <OUTPUTFILE>-o</OUTPUTFILE>
      </CmdParms>
      <Passthru>
        <Cmdlineparm>-h 612 -w 1008</Cmdlineparm>
      </Passthru>
    </ApplicationGroup>

    <ApplicationGroup name='SalesReports'>
      <CmdParms>
        <CARRIAGECONTROL>-x</CARRIAGECONTROL>
        <CODEPAGE>-c</CODEPAGE>
        <OUTPUTFILE>-o</OUTPUTFILE>
      </CmdParms>

      <Application name='Forecasts'>
        <CmdParms>
          <CARRIAGECONTROL>-x</CARRIAGECONTROL>
          <CODEPAGE>-c</CODEPAGE>
          <OUTPUTFILE>-o</OUTPUTFILE>
          <TRC_EXIST>-t</TRC_EXIST>
        </CmdParms>
        <Passthru>
          <Cmdlineparm>-h 1008 -w 612</Cmdlineparm>
        </Passthru>
      </Application>
    </ApplicationGroup>
  </transform>
<Transforms>

```

Figure 9-2 Sample XML with `<ApplicationGroup><Application>` tags

Figure 9-3 shows the transform commands that are generated based on the sample XML and Application Group and Application of the document that is retrieved.

Doc's AG	Doc's Appl	XML Match for AG	XML Match for Appl	Command generated
FinancialReports	Ledgers	Yes	No	c:/opt/txfrm.exe -x A -c 850 -o <outputfilename> -h 612 -w 1008 <datafilename>
SalesReports	Forecasts	Yes	Yes	c:/opt/txfrm.exe -x N -c 500 -t 1 -o <outputfilename> -h 1008 -w 612 <datafilename>
SalesReports	WeeklySummaries	Yes	No	c:/opt/txfrm.exe -x A -c 500 -o <outputfilename> <datafilename>
Accounting	Payable	No	No	c:/opt/txfrm.exe -lm 133 -x M -a 500 -o <outputfilename> -r PDF <datafilename>

Figure 9-3 Table of generated commands

Note: Inheritance is not supported. If an <ApplicationGroup> node is matched, only those options within that node are used for the transform; no parameters that are identified for a parent transform node are used. Similarly, if an <Application> node is matched within an <ApplicationGroup> node, only those options are used for the transform; nothing from the <ApplicationGroup> node is used.

9.2.6 Advanced implementation: Custom Java interface

By using the advanced implementation of the Generic Transform Interface, client developers can write a Java interface to ODWEK that can handle the transform requests in a programmatic way, offering the most application flexibility. Developers can create a class and implement the `transformData()` method to accept document data and details from Content Manager OnDemand and transform the data in any way they choose.

Example 9-3 shows a sample of the `ODTransform.xml` files that can be used in this implementation.

Example 9-3 Sample `ODTransform.xml` file

```
<Transforms>
  <transform>
    <TransformName>MYTXFRM</TransformName>
    <TransformDescription>GENERIC Transform Engine.</TransformDescription>
    <ClientClass>com.companyA.corp.TransformClient</ClientClass>
    <OutputMimeType>application/pdf</OutputMimeType>
    <OutputExtension>pdf</OutputExtension>
    <CmdParms>
      <AG_NAME>agName</AG_NAME>
      <APPL_NAME>applName</APPL_NAME>
      <RECORDFORMAT>recfmt</RECORDFORMAT>
      <RECORDLENGTH>LineLength</RECORDLENGTH>
      <CARRIAGECONTROL>CC</CARRIAGECONTROL>
      <CODEPAGE>CodePage</CODEPAGE>
    </CmdParms>
  </transform>
</Transforms>
```

Similar to the basic implementation, the developer uses this XML stanza to set up the required details for document transformation and how those details are passed to the Java transform interface. Example 9-4 shows an example of how the Java interface can be used with the XML stanza to create a document transform request. The example is a code snippet of how the Client Class that is defined in Example 9-3 might be written to transform data.

Example 9-4 Client Class code snippet for transform data

```
/*
// Testcase: CustomTransform
//
// This class tests the ODWEK Generic Transform's Custom
// Java Interface by implementing the required transformData method.
//
// transformData is called by ODWEK when its corresponding custom
// viewer is called via ODHit.retrieve.
*/
import java.util.*;
```

```

import com.ibm.edms.od.*;

public class CustomTransform {
    public static HashMap transformData(HashMap odMap) throws Exception {
        System.out.println("Inside transformData method");
        // List this transform name from the XML file
        System.out.println(" Transform name: " +
            (String)odMap.get(ODTransform.TXFRM_REQ_NAME));

        // List the property keys and values ODWEK read from the transform XML
        // file and provided to this Custom Class
        System.out.println(" Transform properties:");
        Properties gtProps = (Properties)odMap.get(ODTransform.TXFRM_REQ_PROPS);
        Enumeration<?> enumeration = gtProps.keys();
        List<String> list = new ArrayList<String>();
        while (enumeration.hasMoreElements()) {
            list.add((String)enumeration.nextElement());
        }
        Collections.sort(list);
        for (String key : list)
            System.out.println(String.format("%25s = %-25s", key,
                gtProps.getProperty(key)));

        // Retrieve the native document from ODWEK
        byte[] inDoc = (byte [])odMap.get(ODTransform.TXFRM_REQ_DATA);
        System.out.println(" Native document size: " + (inDoc == null ? null:
            inDoc.length));

        // Retrieve the document resources from ODWEK
        byte[] inRes = (byte [])odMap.get(ODTransform.TXFRM_REQ_RES);
        System.out.println(" Native doc resource size: " + (inRes == null ? null:
            inRes.length));

        // Normally this is where you do the transform or do something with the
        // byte data.
        // Let's just concat the resources if there are any to the doc
        byte[] transformedDoc;
        if (inRes != null) {
            transformedDoc = new byte[inRes.length + inDoc.length];
            System.arraycopy(inRes, 0, transformedDoc, 0, inRes.length);
            System.arraycopy(inDoc, 0, transformedDoc, inRes.length,
                inDoc.length);
        }
        else
            transformedDoc = inDoc;
        System.out.println(" Concatenated resources to doc size: " +
            transformedDoc.length);

        // Send the transformed data back to ODWEK
        HashMap rtnMap = new HashMap();
        rtnMap.put(ODTransform.TXFRM_RESP_DATA, transformedDoc);
        return rtnMap;
    }
}

```

Example 9-4 on page 214 shows how to set up the `HashMap` to pass document byte arrays in and out of this custom interface, and how to define a custom Java class that contains the `transformData()` method.

This code retrieves the raw document data from ODWEK, gathers all of the document details that Content Manager OnDemand might store from loading the data, and then transforms the document data. The transformed document data can be passed back through ODWEK to the original client request.

Table 9-1 lists the *XmITagNames* for the transformation specification.

Table 9-1 XmITagNames for the transform specification

XmITagname	ODConstant	Description
TransformName	TRANSFORM_NAME	Name of the transform. It is used as the viewer argument that is passed to ODWEK Retrieve APIs.
TransformDescription	TRANSFORM_DESC	Description of the transform.
ClientClass	TRANSFORM_CLIENTCLASS	The class name of the custom interface class.
CmdLineExe	TRANSFORM_CMDLINEEXE	Fully qualified name of the transform executable file.
OutputMimeType	TRANSFORM_MIMETYPE	The MIME type of the data as it is returned from the transform.
OutputExtension	TRANSFORM_OUTPUTTEXT	The extension of the data that is returned from the transform.
CmdParms	TRANSFORM_PARMS	The mappings of OD Values to custom variables. See the constant key words that are shown in Table 9-2 on page 216.
Passthru	TRANSFORM_PASSTHRU	These values are passed through ODWEK directly to the transform.
Cmdlineparm	TRANSFORM_PASSTHRU_CMDLINE	These values are passed through ODWEK directly to the transform command line.

Table 9-2 provides information about the XMLTags. These XML tags are used to pass specific values to the transform command line. These XML tags allow the mapping of the command-line option where the specified value can be passed.

Table 9-2 XmITags detailed information

XmITagname	ODConstant	Description
RECORDFORMAT	DOCUMENT_RECORD_FORMAT	The record format of the document as stored in Content Manager OnDemand.
RECORDLENGTH	DOCUMENT_RECORD_LENGTH	The record length of the document as stored in Content Manager OnDemand.
CARRIAGECONTROL	DOCUMENT_CARRIAGE_CONTROL	The carriage control of the document as stored in Content Manager OnDemand.
TRC_EXIST	DOCUMENT_TRC_EXIST	The TRC settings as stored in Content Manager OnDemand.
DOCROTATION	DOCUMENT_ROTATION	The rotation of the document as stored in Content Manager OnDemand.

XmITagname	ODConstant	Description
AG_NAME	AGNAME	The Content Manager OnDemand application group where the document is stored.
APPL_NAME	APPLNAME	The OnDemand application where the document is stored.
CODEPAGE	DOCUMENT_CODEPAGE	The code page of the document as stored in OnDemand.
LINEDELIMITER	DOCUMENT_LINE_DELIMITER	The line delimiter of the document as stored in OnDemand.
INPUTFILE	TXFRM_INPUT_FILE	The Inputfile parameter to be used by the transform.
OUTPUTFILE	TXFRM_OUTPUT_FILE	The OutputFile parameter that is used by the transform.
V9.5 enhancements		
DOCUMENT_CC_ANSI	DOCUMENT_CC_ANSI	Used instead of <CARRIAGECONTROL> to define the command-line option and value when the document's CC Type is "ANSI" as stored in Content Manager OnDemand.
DOCUMENT_CC_MACHINE	DOCUMENT_CC_MACHINE	Used instead of <CARRIAGECONTROL> to define the command-line option and value when the document's CC Type is "Machine" as stored in Content Manager OnDemand.
DOCUMENT_CC_NONE	DOCUMENT_CC_NONE	Used instead of <CARRIAGECONTROL> to define the command-line option and value when the document's CC is "No" as stored in Content Manager OnDemand.
RECORDFORMATFIXED	DOCUMENT_RECORDFORMAT_FIXED	Used instead of <RECORDFORMAT> to define the command-line option and value when the document's RECFM is "Fixed" as stored in Content Manager OnDemand.
RECORDFORMATVARIABLE	DOCUMENT_RECORDFORMAT_VARIABLE	Used instead of <RECORDFORMAT> to define the command-line option and value when the document's RECFM is "Variable" as stored in Content Manager OnDemand.
RECORDFORMATSTREAM	DOCUMENT_RECORDFORMAT_STREAM	Used instead of <RECORDFORMAT> to define the command-line option and value when the document's RECFM is "Stream" as stored in Content Manager OnDemand.
PRMODENONE	DOCUMENT_PRMODENONE	Used instead of <PRMODE> to define the command-line option and value when the document's PRMode is "None" as stored in Content Manager OnDemand.
PRMODESOSI1	DOCUMENT_PRMODESOSI1	Used instead of <PRMODE> to define the command-line option and value when the document's PRMode is "SOSI1" as stored in Content Manager OnDemand.

XmITagname	ODConstant	Description
PRMODESOSI2	DOCUMENT_PRMODESOSI2	Used instead of <PRMODE> to define the command-line option and value when the document's PRMode is "SOSI2" as stored in Content Manager OnDemand.
TRC_YES	DOCUMENT_TRCYES	Used instead of <TRC_EXISTS> to define the command-line option and value when the document's TRC is "Yes" as stored in Content Manager OnDemand.
TRC_NO	DOCUMENT_TRCNO	Used instead of <TRC_EXISTS> to define the command-line option and value when the document's TRC is "No" as stored in Content Manager OnDemand.

Table 9-3 provides information about the OnDemand client HashMap keys that are used for advanced Java implementation.

Table 9-3 OnDemand client hashmap key and descriptions

HashMap key	Description
TXFRM_RESP_DATA	This key is the HashMap key for the transformed data byte[] to be returned to ODWEK.
TXFRM_REQ_NAME	Name of transform for this request.
TXFRM_REQ_METHOD	The method name that is used in the custom Java class. The transformData() method must exist in the client class.
TXFRM_REQ_DATA	The original Content Manager OnDemand Document data that is contained in this request.
TXFRM_REQ_PROPS	The document details as specified or requested in the transform.xml file.



Migration and expiring data and indexes

IBM Content Manager OnDemand (Content Manager OnDemand) provides multiple methodologies for expiring *report* data (documents) and their indexes. In this chapter, we describe the overall lifecycle of report data, including loading, storage, migration, and expiration.

In this chapter, we cover the following topics:

- ▶ Introduction
- ▶ Loading and storing the data
- ▶ Configuring for migration and expiration
- ▶ Reloading data
- ▶ Expiration processing on Multiplatforms and z/OS
- ▶ Expiring data on OnDemand for i

10.1 Introduction

For this chapter, unless explicitly stated otherwise, the term “data” is used to refer to the report data, the extracted documents or segments, and their related indexes and the extracted resources.

A Content Manager OnDemand system logically stores data in *application groups*. An application group is defined by the Content Manager OnDemand administrator. It consists of data that has the same indexing, data storage, and expiration requirements. The application group definition also specifies where the report and document data are stored, how long the data is stored, and how the data expires. The method or methods that can be used to expire the data are a function of the application group parameters that are defined before the data is loaded into Content Manager OnDemand. In a Content Manager OnDemand system, data typically goes through a lifecycle of loading, storing, migration, and an expiration process.

10.2 Loading and storing the data

The Content Manager OnDemand architecture allows the control and management of the data throughout its lifecycle. The data lifecycle begins with running an efficient load process. Each load process invocation ingests report data for a specified application group.

During a load process, Content Manager OnDemand stores report (document) data, its resources, and index data, as shown in Figure 10-1.

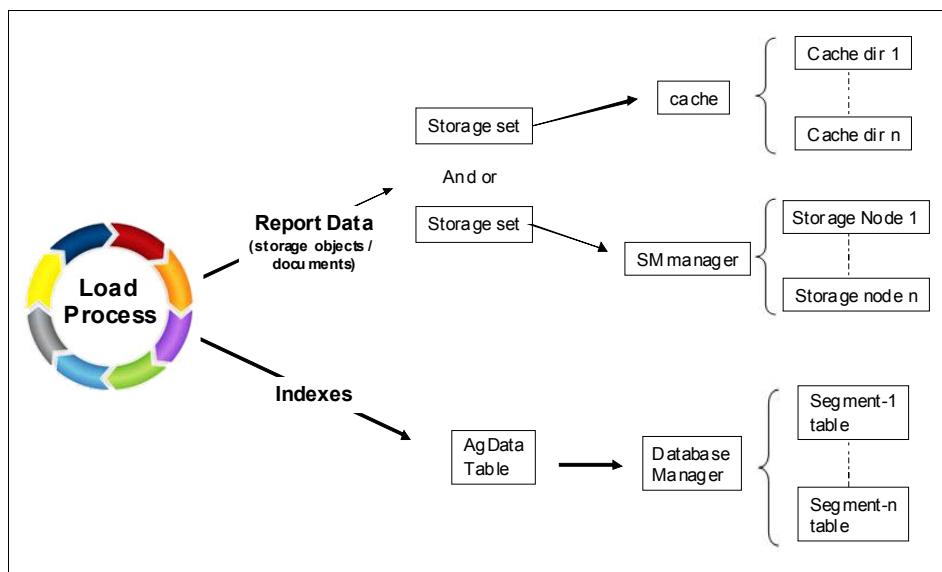


Figure 10-1 Data and index storage locations

The Content Manager OnDemand load process identifies, segments, and compresses groups of documents into storage objects that are then stored in the Content Manager OnDemand archive, as illustrated in Figure 10-1. To improve the efficiency of the storage process, Content Manager OnDemand aggregates the stored documents (typically a few kilobytes in size) into storage objects. This aggregation provides efficient, high-volume storage, retrieval, and expiration performance.

The *object size* is defined by clicking **Advanced** on the Storage Manager tab of the Application Group window. The object size is the size of a storage object in kilobytes (KB). By default, Content Manager OnDemand segments and compresses report data into 10 MB storage objects. For most use cases, the default value is appropriate. Valid values are 1 KB - 150 MB.

Object size value: Exercise caution when you change the object size value. Specifying too large or too small a value can adversely affect performance when you load data.

The storage objects are stored in *storage sets*. The storage sets contain one or more primary *storage nodes*. The storage node points to the location where the data is stored, which can be cache, the storage manager (Tivoli Storage Manager, object access method (OAM), or Archive Storage Manager (ASM)), or a combination.

The primary storage nodes can be on one or more object servers. When the Load Type is Local, Content Manager OnDemand loads data on the server on which the data loading program runs in the primary storage node with the Load Data property specified. If the Load Type is Local, and the storage set contains primary nodes on different object servers, you must select the **Load Data** check box for one primary node on each object server.

The storage set must support the number of days that you plan to maintain reports in the application group. For example, if you must maintain reports in archive storage for seven years, the storage set must identify a storage node (or migration policy on an IBM i server) that is maintained by ASM for seven years.

A detailed description of adding storage sets and storage nodes is in Chapter 5, “Storage management” on page 89 and the related *OnDemand Administrative Guide*.

10.2.1 Storing the report (document) data

To improve efficiency and scalability, stored documents are embedded within storage objects. The storage objects are then stored in cache or a storage manager (OAM, Tivoli Storage Manager, or ASM). The storage objects are eventually expired from the system based on values that are defined by the Content Manager OnDemand administrator. In this section, we describe each scenario and how it is implemented. The parameters that are described in this section are on the Storage Manager tab of the Application Group window unless otherwise specified.

Three sets of data are stored when you load a report:

- ▶ Index data, which is extracted by the indexing program and used by the search process
- ▶ Resources, such as an overlay and fonts, which are used to customize the viewed data
- ▶ Documents (or report segments) that will be viewed

Figure 10-2 on page 222 shows the datasets and illustrates four scenarios of their storage and expiration.

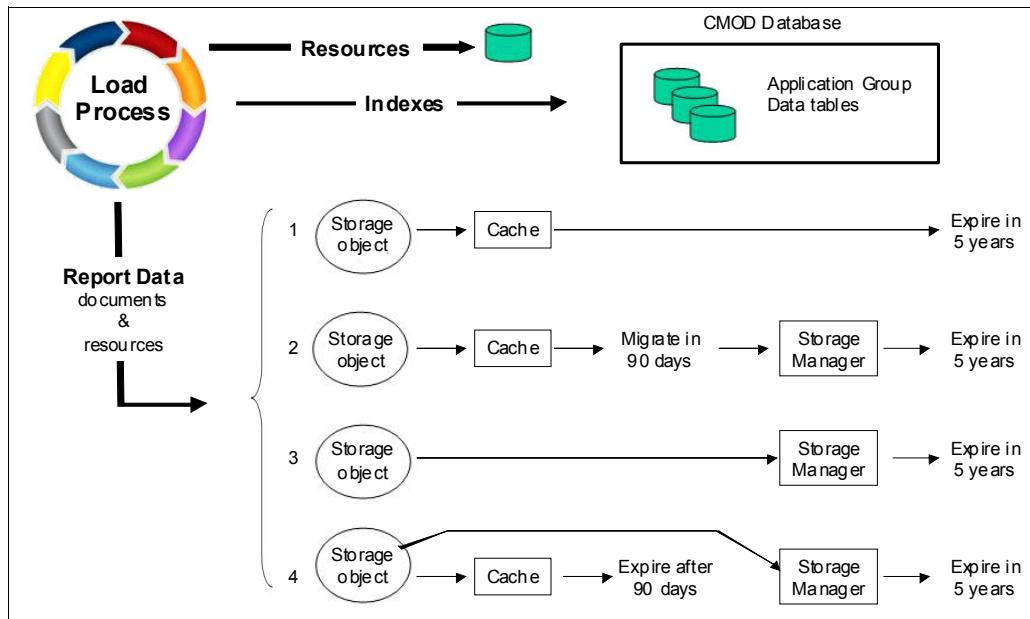


Figure 10-2 Data, resource, and index storage and expiration scenarios

Scenario 1: Cache only, then expiration

In this scenario, the storage object is stored to cache only and it is expired from cache after a predetermined period. Typically, this methodology is employed under the following circumstances:

- The life of the data is short, and hierarchical storage management (HSM) is not necessary.
- The life of the data is long, and a backup process exists for the data in cache.
- The cache device is large enough to hold the total archived data, and the cache device is reliable and performs well.

This method is enabled by selecting a cache-only storage set and entering a number in the Cache Data for __ Days field.

When you select a cache-only storage set, Content Manager OnDemand automatically sets Migrate Data from Cache to No and sets the Expire in __ Days field to the same value as the Cache Data for __ Days field. (The default value is 90 days.)

Selecting a cache-only storage set requires the creation of backup and data management systems that are external to the Content Manager OnDemand system.

Cache-only storage: If the storage set contains cache-only storage nodes, ensure that the Cache Data value and the Life of Data and Indexes value are the same. Otherwise, the add or update operation cannot be completed.

Scenario 2: Cache, then migration to storage, and then expiration

In this scenario, the storage object is first stored to cache for a short period, after which it is migrated to a storage manager for long-term storage.

Typically, this methodology is employed under the following circumstances:

- ▶ Most of the data access occurs during the initial period. After that period, the data is infrequently accessed, if ever. So, after this initial period, the data is migrated to the storage manager.
- ▶ A performance advantage is possible if you retrieve the data from cache versus if you retrieve the data from the storage manager. The performance advantage for cache can occur if the storage manager is on a device that is separate from the Content Manager OnDemand object server, or if the storage manager is local but the storage device is relatively slow, such as tape or an optical disk.

Migrating data from cache

This function, which can be accessed by clicking **Advanced** on the Storage Manager tab of the Application Group window, determines how long the data is kept in cache before it is migrated to archive storage (on a potentially slower archive storage device).

The data needs to be kept on a high-performance storage device for the period during which it is retrieved frequently. The storage set must support the type of media that is required to hold reports that are stored in the application group. For example, if you must maintain reports in cache storage for 90 days and in archive storage for seven years, the storage set must identify a storage node (or migration policy) that causes ASM to maintain the data for seven years, and you must select **Cache Data for __ Days** and enter 90 in its field.

From a user's perspective, no procedural difference exists in retrieving the data from either cache or archive storage. The only user-perceivable difference is the response time. Various archive storage mechanisms provide different performance profiles. For example, when you use OAM and the data is stored in DB2 tables on disk, the response time is as fast as the cache response time. The main difference in response time is based on the type of disk that is used by either method. Conversely, if the OAM data is stored on optical disks or tape, the response time is increased dramatically. If you use a network-attached Tivoli Storage Manager server, the retrieval rates (throughput and response times) are governed by the Tivoli Storage Manager device and the TCP/IP connection to that device.

Typically in a z/OS environment, data is *not* stored in cache. Content Manager OnDemand for z/OS customers typically use OAM as their storage manager. OAM supports storing the data directly in DB2 where the storing and retrieval rates are exceptionally fast, which eliminates the need to maintain and monitor cache file systems in the z/OS file system (zFS) or the hierarchical file system (HFS).

Scenario 3: Storage manager only, then expiration

The storage object is stored directly to the storage manager. Typically, this methodology is employed under the following circumstances:

- ▶ The performance of the storage manager equals the performance of the local file system, which implies that the storage manager stores data to a relatively fast device, such as local disk.
- ▶ Hierarchical storage management is beneficial. An example is z/OS systems where storing directly to OAM is the most popular solution.

If you do not need to maintain reports in cache storage, select a storage set that identifies a storage node (or migration policy) that is maintained by ASM and set Cache Data to No. Content Manager OnDemand automatically sets Migrate Data from Cache to When Data is Loaded.

Scenario 4: Both cache and storage manager, then expiration

The storage object is stored directly to both cache and the storage manager. After a short period, the data is expired from cache. Then, after a much longer period, the data is expired from the storage manager. Typically, this methodology is used under the following circumstances:

- ▶ The cache file system allows more efficient data retrieval.
- ▶ The data needs to be kept for a longer period.
- ▶ The hierarchical storage management (or other features) of the storage manager is required.

The Cache data field determines whether Content Manager OnDemand stores data in cache storage. If the storage set is a cache-only storage set, Yes is the only selection. If the storage set is an archive manager-controlled storage set (OAM, Tivoli Storage Manager, or ASM), you can optionally add storing the data in cache.

Note: Whether the data is stored in cache or in a storage manager, the main performance differences are a result of the following items:

- ▶ The hardware speed (and I/O channels and interfaces) on which the data is stored.
- ▶ The location of the hardware device in relations to the object server.
If the hardware device connects over a TCP/IP link, that link can form a bottleneck, depending on the link's throughput and the required data retrieval rate.

10.2.2 Storing the index data

The Content Manager OnDemand load process extracts document indexes from the report data and stores the indexes in the Content Manager OnDemand database *application group data tables*. With these indexes, users can efficiently locate, select, and retrieve documents. Typically, indexes are expired when the document data is expired.

Each application group is *segmented* into multiple physical tables by using a date or a date and time field. The size of each physical table is determined by the Max rows setting. Each row in the table contains a set of user-defined and system-defined indexes that enable the search for a report segment or a document. Index data is loaded into a table. When the Max rows value is reached, the table is closed and a new table is created. The number of physical tables that represent an application group might grow from 1 to *n*.

10.2.3 Storing the resource data

If data caching is enabled, Content Manager OnDemand stores resources in the cache. Two locations on the Storage Management tab affect how resources are stored:

- ▶ Resource Data
- ▶ Document Data

Resource Data

The following selections are possible for Resource Data:

- ▶ *Always Maintain in Cache*: The resource data stays in cache forever, and it does not expire.
- ▶ *Cache Resource Data for xxx Days*: The resource data stays in cache for xxx number of days before the data expires.
- ▶ *Restore Resources to Cache*: The resource data is not in cache, and the resource data is requested. The resources are restored to cache from the storage manager.

The **ARSLOAD** program saves one copy of a resource on each node for each application group. The resource can be stored multiple times, depending on how the **ARSLOAD** program compares the data. The **ARSLOAD** program compares the last 50 resources against the resource that is generated by the load. If a match is not found, a new resource is stored.

When the **ARSLOAD** program processes a resource group file, it checks the resource identifier to determine whether the resource is present on the system.

If the storage node identifies a client node in OAM or Virtual Storage Access Method (VSAM), the storage manager copies the resources to archive storage.

Document Data

For Document Data, the following selections are valid:

- ▶ *Yes for Cache Data*: You can cache document data and resource data or only resource data.
- ▶ *No Cache*: Document data is not stored in cache.
- ▶ *Cache Document Data for xxx Days*: Document data is stored in cache for xxx number of days before the data expires.

10.3 Configuring for migration and expiration

Many customers choose to expire their document data and indexes somewhere in the range of 5 - 10 years. In one extreme, document and index data might expire daily. In another extreme, document and index data might never expire.

Four typical lifecycle scenarios are common. The Content Manager OnDemand administrator selects the scenario to implement through various parameters (as shown in this section), which are on the Storage Management tab of the Application Group window. The four scenarios are illustrated in Figure 10-2 on page 222.

10.3.1 Migrating index data

Index migration is the process by which Content Manager OnDemand moves index data from the database to archive storage. Index migration optimizes database storage space. With index migration, you can maintain index data for a long time. You typically migrate index data only after users no longer need to access the reports. However, for legal or other requirements, you often must maintain data for a number of months or years.

If a user queries the index data that was migrated, an administrator must act to import a copy of the migrated table or tables by running **ARSADMIN** (Multiplatforms or z/OS) or Start Import into Content Manager OnDemand (**STRIMPOND**) on IBM i. After Content Manager OnDemand maintains the imported index data in the database for the number of days that is specified in the Keep Imported Migrated Indexes field, Content Manager OnDemand deletes the data from the database.

Migration of indexes

This configuration is set up by clicking **Advanced** on the Storage Manager tab of the Application Group window.

This field determines when Content Manager OnDemand migrates index data to archive storage. Choose from *No Migration* or *Migrate After Days*. As a preferred practice, do *not* migrate indexes to archive storage. Indexes that are migrated cannot be searched until after they are imported by an administrator. Use this capability only under limited circumstances.

Closing index tables: Before you can migrate index data, the index tables must be closed. The following Database Organization field options are valid:

- ▶ If the Database Organization field for the application group is set to Single Load per table, this option is no longer supported.
- ▶ If the Database Organization field for the application group is set to Multiple Loads per table, the index table is closed when the *Maximum Rows* value is reached.
- ▶ The Single table for all loads option is available for Content Manager OnDemand for z/OS and Content Manager OnDemand for IBM i. Select the **Single table for all loads** check box if you want to create one database table for each application group. This option is most frequently used when you load a small amount of data. If you select this option, the Maximum Rows field in this window is removed.

To close a table to loading before the Maximum Rows value is reached, run the **ARSTBLSP** program with the **-a1** parameter.

The index data must be migrated only after users no longer need to access the data. If a user must access data in the migrated tables, the process of importing the data into the database requires administrator intervention, and usually results in a significant delay in completing the query. Additional space is required in the database and temporary storage areas to import the data.

To enable the migration of index data, you must define a storage set that identifies a storage node that is maintained by ASM and update the System Migration application group to use the storage set.

10.3.2 Expiring data and indexes

In all four of the storage and expiration scenarios, the index data is stored in the Content Manager OnDemand database in application group data tables. Typically, these indexes are expired when the document data is expired from the system.

Life of Data and Indexes field

This field determines when Content Manager OnDemand deletes documents, resources, and index data from the application group.

The following options are valid:

- ▶ *Never Expires*: Content Manager OnDemand maintains the application group data indefinitely.
- ▶ *Expires in __ Days*: After the data reaches this threshold, Content Manager OnDemand can delete data from the application group the next time that **ARSMINT** (with Content Manager OnDemand for Multiplatforms or z/OS) or Disk Storage Management (DSM) (with IBM i) is run. The default value is 2555 days (seven years). The maximum value that you can use is 99999 days (273 years).

Note: If you plan to maintain application group data in archive storage, the length of time that ASM maintains the data must be equal to or exceed the value that you specify for the Life of Data and Indexes field.

Life of Data and Indexes can be used only if **ARSMINT** (with Multiplatforms or z/OS) or Disk Storage Management (DSM) (with IBM i) handles the expiration.

10.3.3 Expiring document data

Document data expiration is affected by the document expiration type.

Expiration type

The *document expiration type* determines how data is deleted from the application group. The expiration type option is on the Storage Management tab of the Application Group window.

Four expiration types are valid:

- ▶ Load
- ▶ Storage Manager
- ▶ Segment
- ▶ Document

Expiration type: Load

When the expiration type is set to Load, the system deletes an input file (a load) from the application group. Load is the default expiration type. The latest date value from the input file and the Life of Data and Indexes field determine when the data is eligible to be deleted.

Note: The application group *must have* an expiration type of Load if any of the following circumstances are true:

- ▶ You use or plan to use the Enhanced Retention Management feature.
- ▶ You use or plan to use the full text search feature.
- ▶ You use or plan to integrate with the FileNet P8.

For application groups with expiration types of Document, Segment, or Storage Manager, utilities exist to convert these application groups to Load.

Consider engaging IBM Lab Services to provide these services.

With Content Manager OnDemand for Multiplatforms or z/OS, when the expiration type is set to Load, if your object server is on z/OS, and your storage manager is OAM, you can allow OAM to handle the data expiration and Content Manager OnDemand to handle the index expiration by using **ARSEXOAM** program.

With Content Manager OnDemand for i, when the expiration type is set to Load, you can still allow ASM to handle the data and index expiration by creating an expiration level in the migration policy.

Expiration type: Storage Manager (z/OS)

The storage manager (OAM or VSAM) determines when data is deleted from the system. Storage Manager expiration works with either the **ARSEXPIR** program or the **ARSEXE0AM** program.

For more information about how to configure the system to use the **ARSEXPIR** and **ARSEXE0AM** programs, see the *IBM Content Manager OnDemand for z/OS Administration Guide*:

http://www.ibm.com/support/knowledgecenter/SSQHWE_9.0.0/com.ibm.ondemand.administeringzos.doc/aboutpub.htm?cp=SSQHWE_9.0.0%2F7-0

Storage Manager expiration is supported only on Content Manager OnDemand for z/OS systems.

Expiration type: Segment

The system deletes a segment (table) of data from the application group. The system can delete a segment of data only after the segment is closed and every record in the segment reaches its expiration date.

With Multiple Loads per Database Table enabled, the system uses the maximum number of rows to determine when to close a table. A segment likely contains the data from more than one input file. If the Maximum Rows setting is too large, the segment is not expired until all of the documents in the table reach their expiration dates. If the Maximum Rows setting is too small, segments are created constantly and potentially deleted (based on the expiration date). This large number of tables imposes a performance impact during the search query time and expiration time.

The system derives the expiration date from the Segment field (or the date that the data was loaded, if there is no Segment field) and the Life of Data and Indexes field. If the Segment field contains a date in the MMYY format, data is eligible to be deleted on the first day of the month (MM).

To specify the Segment field, complete the following steps:

1. Click the **Field Information** tab.
2. Select a date or date and time field.
3. Select the **Segment** check box.

Expiration type: Document

When the expiration type is set to Document, the system deletes a document from the application group. To determine when to delete a document, the system uses the value of the Expire Date field and the Life of Data and Indexes field. If the Expire Date field contains only the month and year (MMYY format), the system deletes documents on the first day of the month (MM).

To specify the Expire Date field, complete the following steps:

1. Click the **Field Information** tab.
2. Select a date or date and time field.
3. Select the **Expire Date** check box.

Performance note: Individual document deletion is the most costly type of deletion in terms of processor consumption and run time.

10.3.4 Expiring annotations

Annotations for all application groups are kept in a single application group data table, which allows the expiration of annotations to be controlled at a system-wide level. The Life Of Annotations field setup is on the System Parameters General tab. Annotations can be set to never expire or to expire after N days. After the number of days (N) passes and ARSMAINT is run, Content Manager OnDemand removes the annotation.

10.4 Reloading data

If you are migrating data by unloading and then reloading the data, you need to determine your future expiration policy.

Reloading to change the expiration type

For example, if your current expiration policy is set to Storage Manager but you later want to perform holds on the data, during the migration process (when you create the application group and before you load any data), change your expiration policy from Storage Manager to Load.

When you use the Enhanced Retention Management feature with Content Manager OnDemand or IBM Enterprise Records (formerly IBM FileNet Records Manager), Content Manager OnDemand must be in complete control of expiration processing. Therefore, if you are using Tivoli Storage Manager or OAM, you must disable the ability for either of these storage managers to expire data.

Also, you can use Enhanced Retention Management and Content Federation Services for Content Manager OnDemand only with application groups with an expiration type of Load. For those application groups with expiration types of Document, Segment, or Storage Manager, utilities exist to convert these application groups to an expiration type of Load.

Consider engaging IBM Lab Services to provide these services.

Reloading ad hoc stored documents

If you choose not to take advantage of the ability of Content Manager OnDemand to aggregate documents but instead you choose to load documents ad hoc by using the storeDocument Java API, StoreDoc Object Linking and Embedding (OLE) API, or CommonStore, you must migrate the data later.

If you choose not to take advantage of the ability of Content Manager OnDemand to aggregate documents into 10 MB storage objects, this decision might result in millions of small objects that are stored in your storage manager, which might cause the storage manager to experience performance problems when it migrates these small objects to tape.

Note: Consider aggregating these smaller objects into larger objects for performance reasons.

For you to aggregate all of these tiny objects into larger objects after they are stored individually requires that you retrieve and reload them as larger objects. You might want to engage IBM Lab Services to assist you with this task.

Another option is to not migrate objects to tape, but to use another random access hardware device instead.

10.5 Expiration processing on Multiplatforms and z/OS

This section goes into detail about the expiration process on Multiplatforms and z/OS.

10.5.1 Content Manager OnDemand expiration: ARSMAINT

The **ARSMAINT** program manages application group data in the Content Manager OnDemand database and in cache storage.

You typically run the **ARSMAINT** program on a regular schedule to perform the following tasks:

- ▶ Migrate files from cache storage to archive storage.
- ▶ Delete files from cache storage.
- ▶ Optionally, migrate index data from the database to archive storage.
- ▶ Delete index data from the database.

The application group data and the data that you stored in cache are all managed by the **ARSMAINT** program. It is managed by using the storage management values from the application groups that are defined to the system.

Here are the storage management field values that are used:

- ▶ Life of Data and Indexes
- ▶ Length of Time to Cache Data on Magnetic
- ▶ Length of Time Before Copying Cache to Archive Media
- ▶ Length of Time Before Migrating Indexes to Archive Media
- ▶ Length of Time to Maintain Imported Migrated Indexes
- ▶ Expiration Type

10.5.2 Expiring indexes

The **ARSMAINT** program uses the Expiration Type field value to determine how to delete index data from an application group. The **ARSMAINT** program can expire a table of application group data at a time, a load at a time, or individual documents. Ensure that the **ARSMAINT** program command runs periodically (for example, daily) so that Content Manager OnDemand deletes indexes and cache data (and the storage manager deletes archive data, if applicable). By running the **ARSMAINT** program regularly, you ensure that the expired documents can no longer be retrieved.

Additionally, you can start manual expiration processing by running the **ARSMAINT** program from the command line. For example, to run expiration processing, run the following command at the command line:

```
arsmaint -d
```

When the **ARSMAINT** program removes indexes, it saves the following message in the system log:

“128 App1Grp Segment Expire (App1Grp) (Segment)”

One message is saved in the system log for each table that was dropped during expiration processing.

When to run the maintenance processes: Most maintenance processes need to run when no other applications are updating the database or need exclusive access to the database and when you are sure that no one is retrieving documents from the system. For example, you must not perform maintenance on the database while you are loading data into the system.

The relationship between **ARSMAINT** and **ARSSOCKD** processing is illustrated in Figure 10-3.

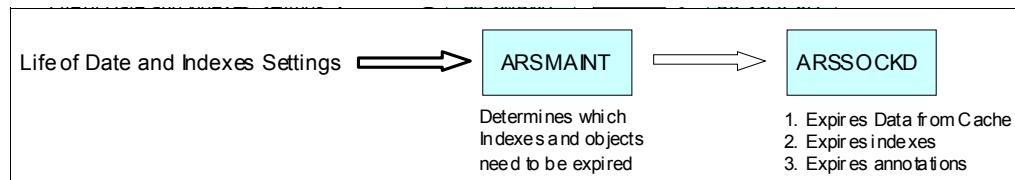


Figure 10-3 Relationship between **ARSMAINT** and **ARSSOCKD** programs

Collecting statistics

Content Manager OnDemand provides two programs to collect statistics on database tables: the **ARSDB** program and the **ARSMAINT** program.

When you run the **ARSMAINT** program to collect statistics, it collects statistics on all of the tables in the database that changed since the last time that you collected statistics. You can automate the collection of statistics by scheduling the **ARSMAINT** program to run with the appropriate options.

You can use the **ARSDB** program to collect statistics on the Content Manager OnDemand system tables. The Content Manager OnDemand system tables include the user table, the group table, and the application group table. For most systems, the Content Manager OnDemand system tables require little maintenance. You can probably schedule the **ARSDB** program to collect statistics once a month (or less often).

The syntax for the **ARSDB** program is shown:

```
/opt/IBM/ondemand/V9.0/bin/arsdb <options>
```

The options are explained:

- | | |
|--|---|
| <ul style="list-style-type: none"> -e -r -s | <ul style="list-style-type: none"> Drop configuration indexes. Create configuration indexes. Collect statistics. |
|--|---|

System log messages

When you run the **ARSMAINT** program, it saves messages about its activities in the system log. The types of messages that are saved in the system log depend on the options that you specify when you run the **ARSMAINT** program.

The number of messages that are saved in the system log each time that expiration processing runs depends on the following factors:

- ▶ The options that you specify for the **ARSMAINT** program
- ▶ The number of application groups that is processed
- ▶ The number of segments of data that is processed
- ▶ The number of cache storage file systems that are defined on the server

Note: You see one set of messages for each object server on which you run the **ARSMaint** program.

For example, when expiration processing starts on a specified server, you might see the following message:

“109 Cache Expiration (Date) (Min%) (Max%) (Server)”

Migration processing uses the specified date (the default is “today” in internal format). Expiration processing begins on each cache file system that exceeds the Max% (default 80%) and ends when the free space that is available in the file system falls below the Min% (default 80%).

One of these messages shows for each storage object that is deleted from cache storage. A storage object is eligible to be deleted when its “Cache Document Data for *n* Days” or “Life of Data” period passes (whichever occurs first).

A storage deletion message looks similar to the following message:

“196 Cache Migration (App1Grp) (ObjName) (Server)”

Also, information-only messages report the percentage of space that is used in the file system.

An information message looks similar to the following message:

“124 Filesystem Statistics (filesystem) (% full) (server)”

Load table (ARSLOAD)

The ARSLOAD table can be used to track loads for expiration. This table maintains a record of all successful loads to application groups with the “expire by load” expiration type.

10.5.3 Removing documents from the Tivoli Storage Manager archive

Removing a document from archive storage means that the backup (if the primary document copy is in cache) or long-term copy (if the primary document copy is in archive) of the document is deleted from the system. You remove documents from archive storage when you no longer have a business or legal requirement to keep them.

A *management class* contains an archive copy group that specifies the criteria that makes a document eligible for deletion. Documents become eligible for deletion under the following conditions:

- ▶ Administrators delete documents from client nodes
- ▶ An archived document exceeds the time criteria in the archive copy group (how long archived copies are kept)

ASM does not delete information about expired documents from its database until expiration processing runs. You can run expiration processing either automatically or manually by command. Ensure that expiration processing runs periodically to allow ASM to reuse storage pool space that is occupied by expired documents.

When expiration processing runs, ASM deletes documents from its database. The storage space that these documents used to occupy then becomes reclaimable. For more information, see “Reclaiming space in storage pools” on page 233.

You control automatic expiration processing by using the expiration processing interval (EXPINTERVAL) in the server options file (dsmserv.opt). You can set the option by editing the dsmserv.opt file. For more information, see the *Content Manager OnDemand Installation and Configuration Guide*:

<http://www.ibm.com/support/knowledgecenter/>

You can obtain more information in the “Running expiration processing automatically” section at the following website:

<http://ibm.co/1i09SdX>

If you use the server option to control when expiration processing occurs, ASM processes expirations each time that you start the server. Afterward, it runs expiration processing at the interval that you specified with the option, which is measured from the start time of the server.

You can manually start expiration processing by running the **EXPIRE INVENTORY** command. Expiration processing then deletes information about expired files from the database. You can schedule this command by running the **DEFINE SCHEDULE** command. If you schedule the **EXPIRE INVENTORY** command, set the expiration interval to 0 (zero) in the server options so that ASM does not run expiration processing when you start the server. You can control how long the expiration process runs by using the **DURATION** parameter with the **EXPIRE INVENTORY** command.

Reclaiming space in storage pools

Space on a storage pool volume becomes reclaimable as documents expire or as they are deleted from the volume. For example, documents become obsolete because of aging.

ASM reclaims the space in storage pools based on a reclamation threshold that you can set for each storage pool. When the percentage of space that can be reclaimed on a volume rises above the reclamation threshold, ASM reclaims the volume. ASM rewrites documents on the volume to other volumes in the storage pool, making the original volume available for new documents.

ASM checks whether reclamation is needed at least once each hour and begins space reclamation for eligible volumes. You can set a reclamation threshold for each storage pool when you define or update the storage pool.

During reclamation, ASM copies the files to volumes in the same storage pool unless you specified a reclamation storage pool. Use a reclamation storage pool to allow automatic reclamation for a storage pool with only one drive. See your ASM documentation for details.

After ASM moves all documents to other volumes, one of the following actions occur for the reclaimed volume:

- ▶ If you explicitly defined the volume to the storage pool, the volume becomes available for reuse by that storage pool.
- ▶ If the volume was acquired as a scratch volume, ASM deletes the volume from its database.

Important: For more information about reclamation processing, including choosing a reclamation threshold, reclaiming volumes in a storage pool with one drive, reclaiming Write Once Read Many (WORM) optical media, reclaiming for copy storage pools, and reclaiming offsite volumes, see your Tivoli Storage Manager documentation.

Managing Tivoli Storage Manager storage

For each automated library, Tivoli Storage Manager tracks in its volume inventory for the library whether a volume has scratch or private status:

- ▶ A *scratch volume* is a labeled volume that is empty or contains no valid data, and it can be used to satisfy any request to mount a scratch volume. To support Content Manager OnDemand, you define scratch volumes to Tivoli Storage Manager. Tivoli Storage Manager uses scratch volumes as needed, and returns the volumes to scratch when they become empty (for example, when all data on the volume expires).
- ▶ A *private volume* is a volume that is in use or owned by an application, and it might contain valid data. Volumes that you define to Tivoli Storage Manager are private volumes. A private volume is used to satisfy only a request to mount that volume by name. When Tivoli Storage Manager uses a scratch volume, it changes the volume's status to private. Tivoli Storage Manager tracks whether defined volumes were originally scratch volumes. Volumes that were originally scratch volumes return to scratch status when they become empty.

Secondary storage of storage volumes

For instructions that describe how to handle physical storage volumes and remove them from the library, see the documentation that is provided by the library manufacturer.

For instructions about documentation that you might need to complete when you remove storage volumes from a library and where to store them for safekeeping, see your organization's media storage guide.

Protecting data with data retention protection

To avoid the accidental erasure or overwriting of critical data, Content Manager OnDemand supports the Tivoli Storage Manager APIs that relate to data retention. *Data retention protection* prohibits the explicit deletion of documents until their specified retention criterion is met. Although documents can no longer be explicitly deleted, they can still expire.

Important notes:

- ▶ Data retention protection is *permanent*. After it is turned on, it cannot be turned off.
- ▶ Content Manager OnDemand does not support deletion on hold data. This feature prevents held data from being deleted until the hold is released.

Tivoli Storage Manager supports two retention policies:

- ▶ In *creation-based retention*, the policy becomes active when the data is stored (created) on the Tivoli Storage Manager server. This policy is the default retention policy method and it is used with normal backup/archive clients.
- ▶ In *event-based retention*, the policy becomes active when the client sends a retention event to the Tivoli Storage Manager server. The retention event can be sent to the server any time after the data is stored on the server. Until the retention event is received, the data is indefinitely stored on the Tivoli Storage Manager server. For Content Manager OnDemand, the retention event is the call to delete the data. A load, unload, application group delete, or expiration of data triggers the retention event.

If you decide to use these policies in Tivoli Storage Manager, the Content Manager OnDemand scenarios that are described in the rest of this section are supported.

Turning off data retention protection

When you turn off data retention protection, the following descriptions explain what happens when you use the creation-based object expiration policy and the event-based retention object expiration policy:

- ▶ Creation-based object expiration policy: Content Manager OnDemand issues a **delete object** command through the Tivoli Storage Manager API. Objects are deleted during the next inventory expiration. If a Content Manager OnDemand application group is deleted, a **delete filesystem** command is issued instead, and the objects are immediately deleted with the file space.
- ▶ Event-based retention object expiration policy: Content Manager OnDemand issues an **event trigger** command through the Tivoli Storage Manager API. The status of the objects that are affected changes from PENDING to STARTED, and the objects are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire. If a Content Manager OnDemand application group is deleted, a **delete filesystem** command is issued instead, and the objects are immediately deleted with the file space.

Turning on data retention protection

When you turn on data retention protection, the following descriptions explain what happens when you use creation-based object expiration policy and event-based retention object expiration policy:

- ▶ Creation-based object expiration policy: Content Manager OnDemand issues no commands to Tivoli Storage Manager. The objects are effectively orphaned by Content Manager OnDemand and are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.
- ▶ Event-based retention object expiration policy: Content Manager OnDemand issues an **event trigger** command through the Tivoli Storage Manager API. The event status of the objects that are affected is changed from PENDING to STARTED, and the affected objects are expired by Tivoli Storage Manager based on their retention parameters. If the retention parameters are set to NOLIMIT, the objects never expire.

If a Content Manager OnDemand application group is deleted, a **delete filesystem** command cannot be used with data retention protection; the operation is treated the same as though a delete is indicated. The status of all of the affected objects is changed from PENDING to STARTED, and the affected objects are expired by Tivoli Storage Manager based on their retention parameters. This action leaves the file space entries in Tivoli Storage Manager, so you must manually delete these entries when the file space is empty (even with data retention protection on).

Recommendations

Consider the following preferred practices when you work with data retention protection:

- ▶ Set up the application groups to expire by load.
- ▶ Define the Tivoli Storage Manager archive copy groups to be event-based, and retain data for 0 days.
- ▶ Run the Tivoli Storage Manager inventory expiration regularly to ensure that expired data is removed.

The following devices are supported by Content Manager OnDemand:

- ▶ IBM DR450 and DR550

These devices are disk-based systems that contain a Tivoli Storage Manager that runs data retention protection.

- ▶ EMC Centera

This device is a disk-based system that is treated as a device by Tivoli Storage Manager. Tivoli Storage Manager must run data retention protection.

10.5.4 Storage Manager-based expiration (z/OS only)

The **ARSEXOAM** and **ARSEXPIR** programs are used for storage manager-based expiration.

ARSEXOAM

The **ARSEXOAM** program is used to process the rows in the ARSOAM_DELETE table that indicate that Content Manager OnDemand OAM objects expired and to remove the associated table entries for those objects. This program works for z/OS only.

Figure 10-4 shows how the **ARSEXOAM** program deletes the index entries for object stores in OAM.

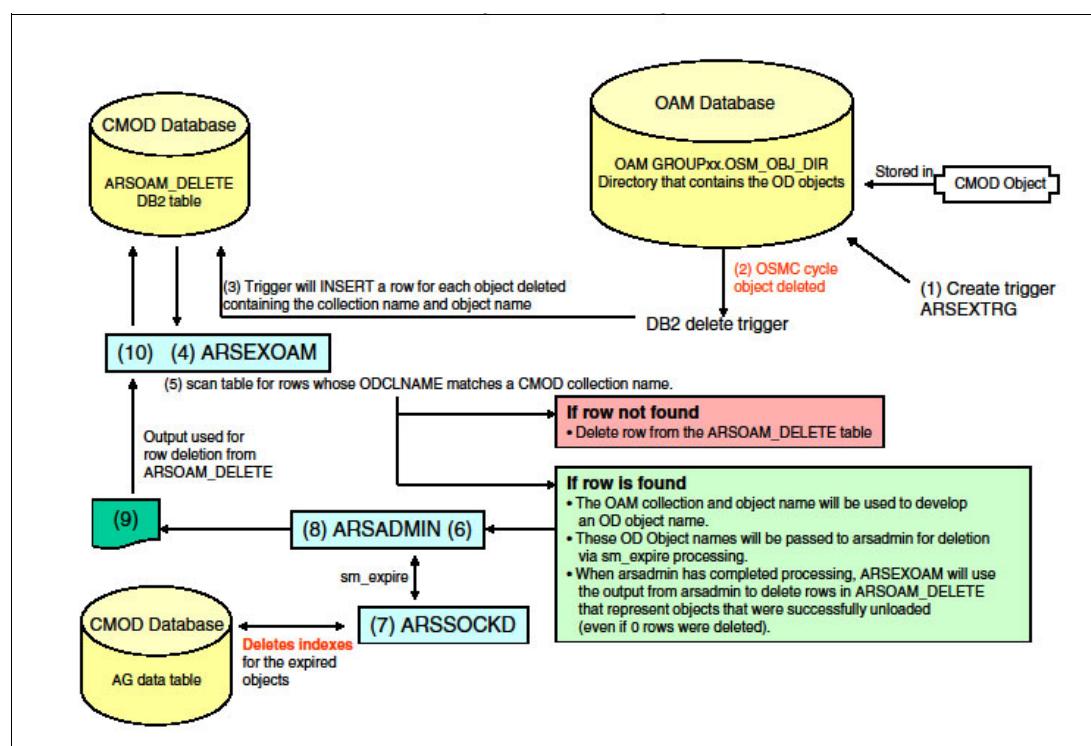


Figure 10-4 How ARSEXOAM deletes index entries for object stores in OAM

Notes:

- ▶ If one object for a load ID is deleted, all of the index entries for that load ID are deleted.
- ▶ Index entries of all OAM objects that are recorded as being deleted by rows in the ARSOAM_DELETE table are deleted regardless of the settings in the Life of Data and Indexes section on the Storage Management tab of the application group.
- ▶ If you plan to use Storage Management expiration, ensure that you set the expiration type of all application groups to Storage Manager.
- ▶ The recommended expiration type for Content Manager OnDemand is Load. Content Manager OnDemand supports the expiration type of Load with the use of **ARSEXOAM** for expiring the indexes in Content Manager OnDemand.
- ▶ Storage Manager expiration is incompatible with Enhanced Retention Manager and Content Federation Services for Content Manager OnDemand.

The following parameters relate to the **ARSEXOAM** program:

▶ **COMMITCNT**

Specifies the number of fetches from the ARSOAM_DELETE, ARSOD, and ARSODIND tables that are performed between COMMITS.

If this parameter is not specified, 1000 is used. If 0 is specified, no commits are performed while fetching. The ARSOD and ARSODIND tables are processed only if Content Manager OnDemand for OS/390 Version 2 migrated index rows are being deleted.

▶ **UNLOADMAX**

Specifies how many objects to hold in memory at any time. The default is 100,000.

▶ **REQLIMIT**

Specifies the maximum number of objects to send to the server in each request. This number defaults to the **ARS_EXPIRE_REQLIMIT** parameter in the ars.cfg, or 100 if

ARS_EXPIRE_REQLIMIT is not specified. Load IDs for the same application group can be grouped up to the **ARS_EXPIRE_REQLIMIT** value. All load IDs in a single expiration request must belong to the same application group. For example, adding
ARS_EXPIRE_REQLIMIT=100 allows up to 100 load IDs for an application group to be processed at a time. The optimum value to use is a function of multiple variables, including table size. Suboptimal values might lead to table scans. EXPLAINS with various SQL that uses the type of SQL that is involved help determine whether an index or a table scan occurs.

ARSEXPIR

The **ARSEXPIR** program can be used to process System Management Facility (SMF) records that indicate that Content Manager OnDemand objects expired and to remove the associated index entries for those objects.

Figure 10-5 on page 238 illustrates two methods that the **ARSEXPIR** program uses to expire OAM and VSAM objects.

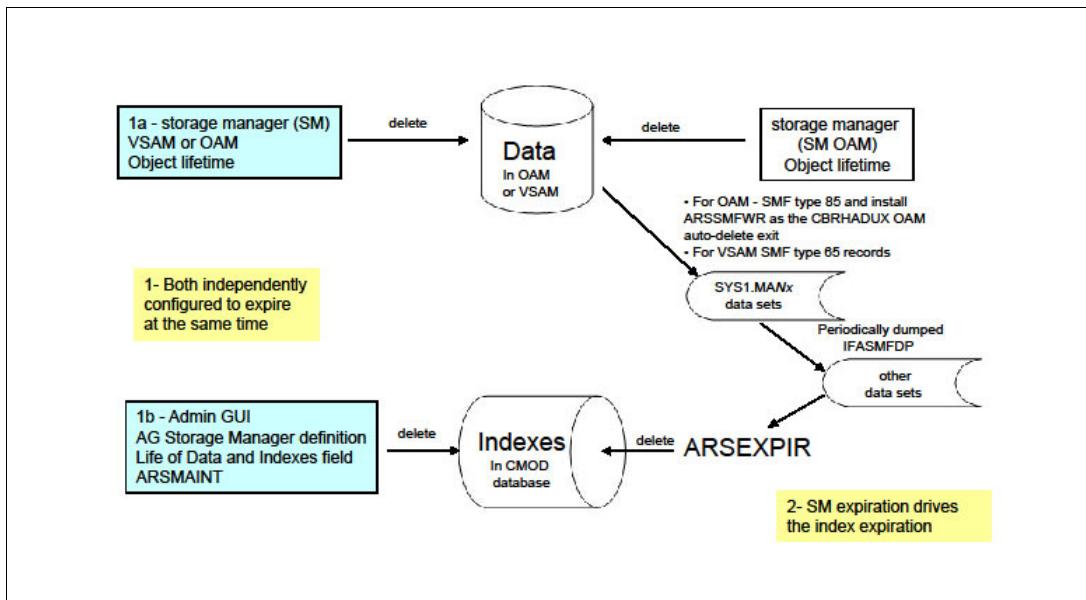


Figure 10-5 Two ways ARSEXPIR expires OAM and VSAM objects

The **ARSEXPIR** program uses SMF type 65 (for VSAM objects) or SMF type 85 (for OAM objects). The installation must collect and install **ARSSMFWR** as the CBRHADUX OAM auto-delete exit. For more information, see “Deleting OAM and VSAM Objects” in the *IBM Content Manager OnDemand for z/OS: Administration Guide*, SC19-1213.

ARSSMFWR determines which objects were deleted. The **ARSEXPIR** program then instructs the Content Manager OnDemand server to remove the index entries.

Notes:

- If one object for a load ID is deleted, all of the index entries for that load ID are deleted.
- Index entries of all objects that are recorded as being deleted by the SMF records are deleted regardless of the settings in the Life of Data and Indexes section on the Storage Management tab of the application group. If you want to use Storage Management expiration, ensure that you set the expiration types of all application groups to Storage Manager.

Important keywords that affect the expiration performance are **COMMITCNT**, **REQLIMIT**, **UNLOADMAX**, and **USERSMF**:

► **COMMITCNT**

This keyword specifies the number of fetches from the ARSOD and ARSODIND table that are to be performed between COMMITS. If this number is not specified, 1000 is used. If this number is 0, no commits are performed while fetching. This parameter is used only if Content Manager OnDemand for OS/390 Version 2 migrated index rows are being deleted.

► **REQLIMIT**

This keyword specifies the maximum number of objects to send to the server in each request. The **REQLIMIT** keyword defaults to the **ARS_EXPIRE_REQLIMIT** parameter in the **ars.cfg**, or 100 if **ARS_EXPIRE_REQLIMIT** is not specified.

► **UNLOADMAX**

Specifies how many objects to hold in memory at any one time. The default is 100,000.

► **USERSMF**

This keyword specifies the SMF record type that is written by the **ARSSMFWR** exit (if used). This parameter can be omitted if **ARSSMFWR** is omitted. For more information about the **ARSSMFWR** exit, see *IBM Content Manager OnDemand for z/OS Configuration Guide*, SC19-3363.

10.6 Expiring data on Content Manager OnDemand for i

In most circumstances, you must run Disk Storage Management (DSM) and Archived Storage Management (ASM) to expire data from Content Manager OnDemand for i.

10.6.1 Content Manager OnDemand expiration

Disk Storage Management (DSM) is the process for performing Content Manager OnDemand based expiration. DSM performs the following functions:

- Controls the expiration of indexes and data from Content Manager OnDemand (if you do not use storage manager-based expiration).
- Migrates data from cache to the storage manager (if the Migrate Data from Cache option is not set to When data is loaded).
- Expires data from cache if Cache Data is set to Yes.

If you do not run DSM, your disk storage requirements for Content Manager OnDemand might be higher than expected. The number of objects that are stored in the integrated file system (IFS) might also be higher than necessary, which results in longer save and restore times.

Note: If you have never run DSM, the first execution of the Start Disk Storage Management (**STRDSMOND**) command might last for an extended period.

If you want to configure Content Manager OnDemand so that DSM is not required in the future, see the section “Eliminating the need to run Disk Storage Manager (DSM)” in the latest *Content Manager OnDemand for i Common Server Administration Guide*, SC19-2792.

10.6.2 Storage Manager expiration

ASM is the process for performing Storage Manager-based expiration. ASM performs the following functions:

- ▶ Controls the expiration of indexes and data from Content Manager OnDemand (if you use Storage Manager-based expiration)
- ▶ Aggregates data before it migrates it to archive media (if you select the Aggregation option in the migration policy)
- ▶ Moves data between storage levels of the migration policy

If you do not run ASM, your disk storage requirements for Content Manager OnDemand are probably higher than expected. The number of objects that are stored in the IFS is also higher than necessary, which results in longer save and restore times.

If you never run ASM, the first execution of the Start Archived Storage Management (**STRASMOND**) command or the Start Disk Storage Management (**STRDSMOND**) command with the **STRASMOND** parameter set to YES might last for an extended period.

For more information about expiring archives by using ASM, see *Expiration processing in Common Server Archive Storage Manager (ASM)*:

<http://www.ibm.com/support/docview.wss?uid=swg21317082>



Exits

In IBM Content Manager OnDemand (Content Manager OnDemand), you can use exit points to customize and enhance the standard functionality within the product. This chapter introduces various exit points within the Content Manager OnDemand product. By using working sample code, we present examples of the types of operations and enhanced functions that are possible.

In this chapter, we cover the following topics:

- ▶ Introduction to user exits
- ▶ ACIF exits
- ▶ OS/390 indexer exits
- ▶ System administration
- ▶ Customized functions (Multiplatforms and z/OS only)

11.1 Introduction to user exits

A *user exit* is a point during processing where you can run a user-written program and return the control of processing after your user-written program ends. Several kinds of exits exist. In this chapter, we describe the exits based on the following categories:

- ▶ AFP Conversion and Indexing Facility (ACIF) indexing

The ACIF indexer contains user exit points for increased flexibility. Four exit points are available during ACIF processing where user programs can be configured:

- Input record exit
- Index record exit
- Output record exit
- Resource exit

- ▶ OS/390 indexer exits

The OS/390 indexer supports three exits to assist with indexing and loading documents into Content Manager OnDemand:

- Anystore exit
- Input exit
- Index exit

- ▶ System administration:

- System log exit
- Print exit

- ▶ Customized functions:

- User exit
- Load exit
- Client Retrieval Preview exit
- Report specifications archive definition exit
- Table space create exit
- **ARSYSPIN** and sample **APKACIF** exit on z/OS

Content Manager OnDemand provides data at each exit that can serve as input to the user-written programs. By using these exits, you can perform functions, such as sending emails based on events in the system, updating index values through a print request, and cleaning up data as it is loaded into Content Manager OnDemand. Unlimited possibilities are available with the Content Manager OnDemand exits. We provide samples to act as a guide for creating customized user exit programs.

Important: Always recompile all of the customized user exits after you upgrade the Content Manager OnDemand software because the header files might change with different versions.

PDF Indexer: The PDF Indexer does not support any user exits.

11.2 ACIF exits

The ACIF user exit is a point during the ACIF processing where control is transferred from ACIF to a user-written program. After the user-written program finishes, the control is returned to ACIF. User programs can be configured at four points during ACIF processing: input, indexing, output, and resource.

Note: ACIF exits are called for every input, indexing, output, and resource record. ACIF exits are not limited to being called only one time for each file.

In Multiplatforms, ACIF user exits must be written in C. In z/OS, ACIF user exits can be written in C, COBOL, or assembler. For more information, see the ‘Special considerations for APKACIF exits written in COBOL’ section in the *IBM Content Manager OnDemand for z/OS, V9.0, Administration Guide*, SC19-3364. ACIF exits do not exist in Content Manager OnDemand for IBM i.

For detailed documentation about each exit point, see *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC19-3354, and *IBM Content Manager OnDemand for z/OS and OS/390 - Indexing Reference*, SC27-1375.

11.2.1 New macro for user exits

Because the default installation directory changed for Content Manager OnDemand V9, the **arsload** program supports a new macro to make user exits more portable.

For example, instead of specifying the exit as
INPEXIT=/opt/IBM/ondemand/V9.0/exits/acif/asciinpe, specify the following items in the indexing parameters:

```
INPEXIT=$(OD_ACIF_EXIT_DIR)asciinpe
```

The **arsload** program substitutes the correct path for the platforms.

This macro works for all four ACIF user exits. The macro is not supported if ACIF is run outside of the **arsload** program.

11.2.2 Input record exit

ACIF provides the input record exit so that you can add, delete, or modify records in the input file before they are processed by ACIF. The primary purpose of this exit is to modify input records before ACIF accesses the records. The exit program is started by the ACIF **inpexit** parameter.

The input exit can be used to insert indexing information. More common uses are to remove null characters, truncate records, add carriage control, and change code pages. In general, indexer parameters need to reflect what the input record looks like *after* the input exit runs. The only exception is the **FILEFORMAT** indexer parameter, which needs to correspond to the input record *before* it is passed to the input exit. For example, if the file contains ASCII data and uses the ASCII stream delimiter x'0A', specify (NEWLINE=x'0A'), not (NEWLINE=x'25'), even if you use the **apka2e** exit to convert ASCII to EBCDIC. Otherwise, ACIF does not pass the correct record to the **apka2e** input exit.

Content Manager OnDemand provides three input record exits:

- ▶ **apka2e**
- ▶ **asciinp**
- ▶ **asciinpe**

You can either use these input record exits as samples to build from, or you can compile them and run them as is. These programs are documented in *IBM Content Manager OnDemand for Multiplatforms - Indexing Reference*, SC18-9235, and are described briefly in the following sections.

The apka2e exit

The **apka2e** exit translates data that is encoded in ASCII (code set IBM-850) into EBCDIC (code set IBM-037). *If you are converting line data to AFP, consider converting the data to EBCDIC.* A much wider selection of EBCDIC coded fonts is available than ASCII fonts. Many customers find that it is easier to use character sets and code pages that are supplied by IBM than to create their own character sets and code pages. To use these predefined EBCDIC coded fonts, the data must be in EBCDIC.

When you use the **apka2e** exit, you must manually change your indexing parameters:

- ▶ Change an ASCII CPGID to an EBCDIC CPGID; for example, change CPGID=850 to CPGID=500.
- ▶ Change the HEX codes for the triggers and index names from ASCII to EBCDIC. If you do not, you receive ACIF return code 16, which states that it cannot find trigger1 or any fields.

We used a hex editor to determine the new EBCDIC values and typed them by keyboard edit into the parameter file. If you do not have a hex editor, you can find conversion tables on the Internet.

For more information about how to update indexing parameters, see 11.2.6, “Debugging input user exit programs” on page 247.

The asciinp exit

The **asciinp** exit program is used when the data does not contain carriage controls. Instead, it contains “PC style” carriage returns and form feeds X'0D0A' and X'0D0C'. This program is provided by IBM. The program transforms the ASCII data stream into a format that contains a carriage control character in byte 0 of every record.

The **asciinp** exit performs the following actions:

- ▶ Inserts a new page command (X'31') at the top of the first page.
- ▶ Removes the ASCII carriage return (X'0D').
- ▶ Inserts an ASCII new line (X'20') carriage control at byte 0 of each line, except the first line on a new page.
- ▶ Replaces the ASCII form feed (X'0C') with an ASCII new page command (X'31').
- ▶ Leaves X'0A' in the file.

Note: Because the **asciinp** exit inserts carriage control characters in byte 0 of your document, and leaves X'0A', it changes the position of the triggers and fields. If you use this exit, you must add 1 to the column offsets for the triggers and fields.

The asciinpe exit

The **asciinpe** exit combines the previous two exits. It converts the data from ASCII to EBCDIC and inserts EBCDIC carriage control characters. For full documentation about this sample program, see the `asciinpe.c` source code.

11.2.3 Index record exit

Use the index record exit to modify or ignore the records that ACIF writes in the index object file. The program, which is specified in the ACIF *idxexit parameter*, receives control before a record is written to the index object file. The user-written program can instruct ACIF to use the record, to not use the record, or to edit the record before the record is inserted into the index object file.

A good use of this program is for an application that must pull an index from a source other than the document. The application group can be set up with a default index; then, the user exit program can grab the appropriate index from this secondary source and replace the default value that was in the index record. The record is then sent back to ACIF.

Another example is to modify the format of an existing index. Example 11-1 shows a sample index exit C program to update the date format from mmddyy to mm/dd/yy.

Important: The ACIF index file is in AFP format. It is important to understand the structure of AFP before you write or modify an index exit.

Example 11-1 Sample ACIF index exit program

```
#define _c_APKIND
/*********************************************************/
/*
 * MODULE NAME: UPDDATE.C
 */
/*
 * SYNOPSIS: ACIF Sample Index Exit
 */
/*
 */
/*
 * DESCRIPTION: This module converts the date format
 *               from mmddyy to mm/dd/yy before adding the
 *               record to the index object file
 */
/*********************************************************/
#include "apkexits.h"/* standard acif exit header file */

long
INDEXEXIT( INDEXEXIT_PARMS *exitstruc )
{
    int i;

    if ( exitstruc->eof != IDX_EOFLAG )
    {
/*********************************************************/
/* Look for TLE with attribute name "mmddyy"      */
/*********************************************************/
        if (
            (exitstruc->record[13] == 0x6D) &&
            (exitstruc->record[14] == 0x6D) &&
            (exitstruc->record[15] == 0x64) &&
            (exitstruc->record[16] == 0x64) &&
            (exitstruc->record[17] == 0x79) &&
            (exitstruc->record[18] == 0x79))
        {
/*********************************************************/
/* TLE length is now 40 (was 38)                  */
/*********************************************************/
            exitstruc->record[ 2 ] = 0x28;

/*********************************************************/
/* Attribute value count is now 12 (was 10)      */
/*********************************************************/

            exitstruc->record[19] = 0x0C;

```

```

/*****************************************/
/* Relocate attribute qualifier triplet X'80' */
/*****************************************/
for (i=40; i>30; i--)
    exitstruc->record[i] = exitstruc->record[i-2];

/*****************************************/
/* Change mmddyy to mm/dd/yy */
/*****************************************/
exitstruc->record[30] = exitstruc->record[28];
exitstruc->record[29] = exitstruc->record[27];
exitstruc->record[28] = 0x61;
exitstruc->record[27] = exitstruc->record[26];
exitstruc->record[26] = exitstruc->record[25];
exitstruc->record[25] = 0x61;

/*****************************************/
/* record length has increased to 41 (was 39) */
/*****************************************/

exitstruc->recordln = 41;

}

exitstruc->request = IDX_USE;
}

return( 0 );
}

```

11.2.4 Output record exit

Use the output record exit to modify or ignore the records that ACIF writes to the output document file. The program is started by the ACIF **outexit** parameter, and it gives control to the user program before a record (which can be a Structured Field or can be line data) is written to the output (.out) file.

Example 11-2 shows a sample output exit program that deletes records from the output file. This program checks each Structured Field to determine whether it is an AFP record. If the record does not begin with Hex 5A, the exit program instructs ACIF not to use this record.

Important: The ACIF output file can be in either Line Data or AFP format. If the ACIF output file is in AFP format, it is important to understand the structure of AFP before you write or modify an output exit.

Example 11-2 Sample ACIF output exit program

```
#define _c_ACCT_OUT
/***************************************************************/
/*
 * MODULE NAME: ACCT_OUT.C
 */
/*
 */
/* SYNOPSIS: ACIF Output Exit
 */
```

```

/*
/* DESCRIPTION: This program will delete all non- AFP records (or      */
/*                records that do not begin with X(5A) from the          */
/*                output object before giving control back to ACIF       */
/*                */                                                 */
/***************************************************************/
/* Standard acif exit header file           */
/***************************************************************/
*/
#include "apkexits.h"

long
OUTEXIT( OUTEXIT_PARMS *exitstruc )
{

/***************************************************************/
/* Delete all records from the output that do not begin with Hex '5A' */
/***************************************************************/

    if(exitstruc->eof != ACIF_EOF)
    {
        if(exitstruc->record[0] == 0x5A)
            exitstruc->request = ACIF_USE;
        else
            exitstruc->request = ACIF_DELETE;
    }
    return( 0 );
}

```

11.2.5 Resource exit

If you want to prevent ACIF from collecting a specific type of resource, such as overlays, you can use the ACIF **restype** parameter. However, if you want to prevent ACIF from writing a specific resource to the resource file, use the resource exit.

The resource exit is best used to control resources at the file name level. For example, you want to exclude particular fonts from the resource file. You can code this exit program to contain a table of the fonts that you want to exclude and filter them from the resource file. The program that is invoked at this exit is defined in the ACIF **resexit** parameter.

ACIF does not start the exit for the following resource types:

- ▶ Page definitions: The pagedef is a required resource for converting line data to AFP and it is never included in the resource file.
- ▶ Form definitions: The formdef is a required resource for processing print files. If you do not want the formdef to be included in the resource file, specify **restype=none** or explicitly exclude the formdef from the **restype** list.
- ▶ Coded fonts: If you specify MCF2REF=CF, ACIF writes coded fonts to the resource file if they are included in the **restype** list. The default is MCF2REF=CPCS; therefore, ACIF does not write coded fonts to the resource file.

11.2.6 Debugging input user exit programs

When you work with an input exit, you must know how the exit changed your data before you load it. A way to determine how the exit changed the data is to set up ACIF to run in stand-alone mode (not called from **arsload**).

To set up ACIF to run in stand-alone mode, create an indexing parameter file with no triggers, fields, or indexes that are defined, and use the value CONVERT=NO. Include your input file, output file, and the input exit routine in the parameter file. Then, run **arsacif** from a command line while you point to this parameter file. Example 11-3 shows our ACIF parameter file, `parmfile`. Use the following command to run stand-alone ACIF:

```
arsacif parmdd=parmfile
```

This command starts ACIF with the user exit, runs the exit, and writes the output to the file that is specified by the **OUTPUTDD** parameter. You can inspect the output file to ensure that the exit did what you expected. You can also use this output file in the graphical indexer to index your post-exit file because the exit routine might change the location of your triggers and fields.

Example 11-3 ACIF parameter file

```
CC=YES
CCTYPE=Z
CONVERT=NO
CPGID=850
FILEFORMAT=STREAM,(NEWLINE=X'0A')
INPUTDD=C:\temp\billing_input.txt
OUTPUTDD=C:\temp\billing_input.txt.out
RESTYPE=NONE
INPEXIT=C:\Program Files\IBM\OnDemand for Windows\V9.0\exits\acif\asciinp.dll
```

Important: Specify the *complete path* in the **inpexit**, **idxexit**, **resexit**, or **outexit** parameters. Nothing is more frustrating than trying to debug an exit that is never called because another exit with the same name is started because of the **PATH** environment variable.

Another method is to run **arsload** with the **-i** option, which runs only the indexer and does not load any files. In this case, it is not necessary to add **INPUTDD** or **OUTPUTDD** to the indexing parameters in the application. Running **arsload** with the **-i** option creates the **.ind** and **.out** files, and leaves them on the system for you to view.

11.3 OS/390 indexer exits

The OS/390 indexer can be used to extract index data from and generate index data about line data and AFP reports. In addition, other data types, such as TIFF images, can be captured by using the **anystore** exit. The OS/390 indexer is available on the z/OS platform for all versions and on the AIX platform beginning with Content Manager OnDemand V9.0. The OS/390 indexer supports the following exits to assist with indexing and loading documents into Content Manager OnDemand:

- ▶ The ANYSTORE exit (**ANYEXIT**) can be used to capture any type of data. The exit is responsible for reading the load file and returning the index values and documents to the indexer. A sample exit is provided that loads TIFF images.
- ▶ The INPUT exit (**INPEXIT**) can be used with line print data. It allows the load file contents to be modified by the exit before they are stored in Content Manager OnDemand.
- ▶ The INDEX exit (**INDEXEXIT**) can be used with any data type. It allows the index values for a document to be modified before they are stored in Content Manager OnDemand.

11.3.1 ANYEXIT exit

ANYEXIT is the anystore exit, which captures any type of data. The exit reads the data to be captured, breaks it into documents, and determines the index values. The sample anystore exit captures TIFF images by using a pre-generated set of indexing instructions that are read from a separate file.

The anystore batch capture exit can be used to provide all segment and index data to the report capture program. This exit program is called from the report capture process.

The exit is called dynamically during the capture process. The capture program calls the exit when the indexing instructions for the application include the **ANYEXIT** parameter. The report administrator provides a program name for the anystore exit.

The report capture program expects the anystore exit to pass back all segment data and the associated index information. The capture program performs only the data management functions that are required for the capture process (document compression, document store, index management and store, and so on).

A sample COBOL exit is provided in ARSEXANY, which is in member SARSINST(ARSEXANY) with the COBOL copybook ARSANYBK. A sample C exit is provided in ARSECANY, which is in member SARSINST(ARSECANY) with the C header files ARSANYBH and ARSZ390H.

11.3.2 INPEXIT exit

INPEXIT is the input exit, which is provided to allow more processing of the report input before the report is stored. This exit can be used only when the **INDEXSTYLE** is *not* set to AFP and when the **ANYEXIT** is *not* specified. The exit is called dynamically during the report capture process. The report capture routine calls the exit when the indexing parameters specify an input exit name in the **INPEXIT** parameter.

The report administrator provides a program name for this parameter.

No restrictions exist for the type of processing that can be performed in an input exit, except that the exit must pass the standard parameter list back to the report capture program. Values must be supplied for all parameters.

Beginning with Content Manager OnDemand for z/OS V8.4 or later, a line print file can have a fixed record length that is greater than 512 or a variable record length. To support this capability, a new parameter format is provided. The old parameter format is still supported for compatibility with earlier versions.

To learn more about how to create an input exit, the details of the new parameter format, and how Content Manager OnDemand determines whether to use the old or new parameter format, see *IBM Content Manager OnDemand Version 9 Release 5 - Indexing Reference*, SC19-3354.

11.3.3 IDXEXIT exit

IDXEXIT is the index exit, which is provided to allow the report indexes to be modified before insertion into the application group data table. This exit can be used with any type of report that is captured by the OS/390 indexer. The exit is called dynamically during the capture process. The capture program calls the exit when the indexing instructions for the application include the **IDXEXIT** parameter.

The report administrator provides a program name for the index exit.

No restrictions exist for the type of processing that can be performed in an index exit, except that the exit must pass the standard parameter list back to the capture program. A sample COBOL exit is provided in ARSEXNDX, with the COBOL copybook ARSINDBK. A sample C exit is provided in ARSECNDX with the C header file ARSINDBH.

For more information about the OS/390 indexer, see *IBM Content Manager OnDemand for z/OS and OS/390 - Indexing Reference*, SC27-1375.

11.4 System administration

In this section, we describe exits that are used for system administration: system message logging and server printer configuration. These exits are in the bin directory of the Content Manager OnDemand installation.

11.4.1 System log exit for Multiplatforms

The Content Manager OnDemand system log is a tool that is used by administrators. You can use the Content Manager OnDemand Administrator Client to configure Content Manager OnDemand to record information, warning, and error messages in the system log. Content Manager OnDemand records messages, such as when users log on and log off the system. Content Manager OnDemand also records messages for application group activity, such as when clients query and retrieve data. Each operation that is performed by a user that involves a connection to the Content Manager OnDemand server can be logged. The detail that is captured within the system log can be configured so that only certain messages are retained, and other messages can be discarded.

In addition, you can configure Content Manager OnDemand to send these messages to the **arslog** exit. The system log exit is supplied in the arslog file that is in the bin directory of the Content Manager OnDemand installation root for each platform. If the arslog file is opened in a text editor, it contains comments that provide a brief description of the exit and the order of the parameters that Content Manager OnDemand supplies to this exit. By default, the system log exit is not initialized within Content Manager OnDemand. Therefore, if you edit the arslog file to capture information, the exit does not run automatically.

To activate the system log exit, complete the following steps:

1. Start the Administrator Client and log on to the server on which you intend to use the system logging exit.
2. Right-click the name of the server in the list and select **System Parameters**, as shown in Figure 11-1 on page 251.

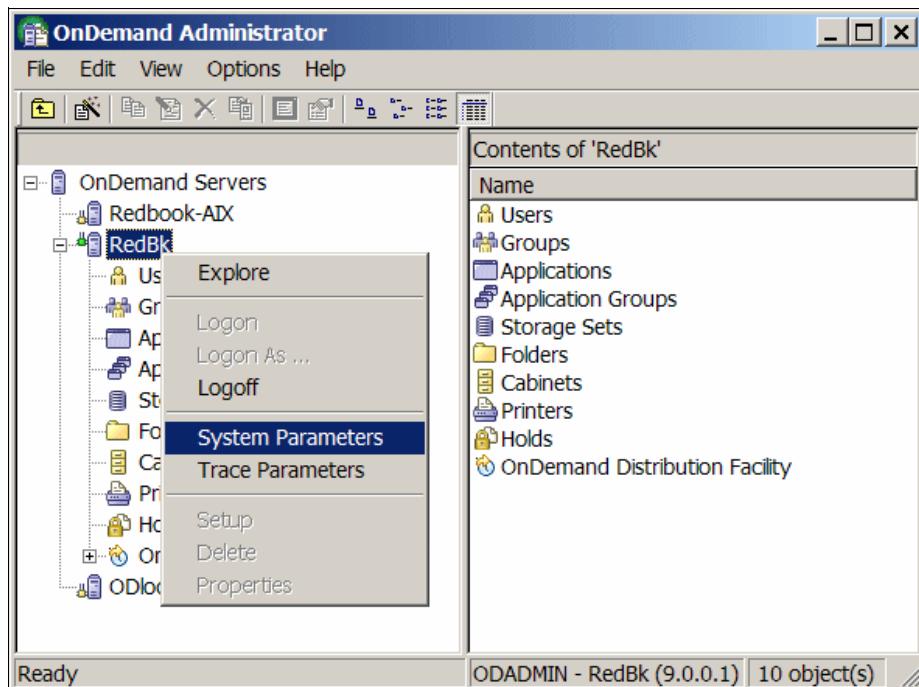


Figure 11-1 Select Content Manager OnDemand system parameters

3. To choose a User Exit Logging option, select the option.

Tip: The `arslog` exit file is run by the same user that owns the `arssockd` process that calls this exit. A common reason for receiving no response from this exit is *access permissions* on either the `arslog` file itself or files and directories that are accessed within `arslog`.

Content Manager OnDemand provides an exit for each of the four system logging event points. Use these exits to filter the messages and act when a particular event occurs. For example, you can provide a user exit program that sends a message to a security administrator when an unsuccessful logon attempt occurs.

System log exit samples

To demonstrate the common uses for the system log exit, we provide two typical examples:

- ▶ Capturing failed logon attempts (AIX)
- ▶ Notifying another system when a load completes (AIX)

For simplicity, we do not demonstrate the system log exits across all supported platforms. We recognize that the scripting languages between platforms vary, but the principles that we describe here are uniform across all supported platforms; only the syntax differs.

Capturing failed logon attempts (AIX)

Example 11-4 on page 252 is an extract from a simple system logging exit that captures *message code 31* (a failed logon attempt) and writes the user ID that was used and information about the network address of this user to a file. In this case, the file name is a combination of the system date and the string `failedlogon.log`. This system log exit writes all of the failed logon attempts for each day to a file that can then be sorted and analyzed by other utilities to alert for possible security risks.

Example 11-4 Capturing failed logon attempts (AIX)

```
# $1 - OnDemand Instance Name
# $2 - Time Stamp
# $3 - Log Identifier
# $4 - Userid
# $5 - Account
# $6 - Severity
# $7 - Message Number
# $8 - Message Text
#
case $7 in
    31) echo $4 $8 >> /home/archive/`date +"%d-%m-%Y"`${failedlogon}.log;;
        *) echo $@ > /dev/null;;
esac

exit 0
```

For the exit sample that is provided in Example 11-4, we also provided a small sample of what the output of this exit might look like (Example 11-5). For example, you can see in the output that several unsuccessful attempts were made from the same machine and different user IDs were used for each attempt. In this example, by adding parameter 2 (\$2) to the output and resorting the file, we can further establish the times of these attempts.

Example 11-5 Sample exit output

```
ADMIN Failed login: GB55102K3.myServer.ibm.com 9.9.9.9
MARTIN Failed login: GB55102K3.myServer.ibm.com 9.9.9.9
FRED Failed login: GB55102K3.myServer.ibm.com 9.9.9.9
USER1 Failed login: GB55102K3.myServer.ibm.com 9.9.9.9
USER2 Failed login: GB55102K3.myServer.ibm.com 9.9.9.9
```

Notifying another system when a load completes (AIX)

This sample is used in a production environment where the number of load jobs that are sent to Content Manager OnDemand must be controlled so that the next load job is sent only when the previous load job completed successfully. We use this exit in this example because a limited amount of disk space is available in the location on the Content Manager OnDemand server where the load files are received from the remote machine. And, the load files are large.

Example 11-6 shows how the exit collects virtually all of the available information when it receives *message number 87* (a successful load). This information is then used as the input for another script, which notifies the remote machine that the load is complete and the next report file can be sent.

Example 11-6 Controlling load jobs (AIX)

```
# $1 - OnDemand Instance Name
# $2 - Time Stamp
# $3 - Log Identifier
# $4 - Userid
# $5 - Account
# $6 - Severity
# $7 - Message Number
# $8 - Message Text
#
```

```

# if [ $6 = "3" ]; then
#   print $@ >> /home/archive/InfoMsg.log
# fi

case $7 in
#
#      msg num 87 is a successful load
#
87) echo "Instance : $1" >> /arsacif/companyx/arslog.out
     echo "Time Stamp : $2" >> /arsacif/companyx/arslog.out
     echo "Log Identifier : $3" >> /arsacif/companyx/arslog.out
     echo "Userid : $4" >> /arsacif/companyx/arslog.out
     echo "Account : $5" >> /arsacif/companyx/arslog.out
     echo "Severity: $6" >> /arsacif/companyx/arslog.out
     echo "Message Number: $7" >> /arsacif/companyx/arslog.out
     echo "Message Text : $8" >> /arsacif/companyx/arslog.out
     /arsacif/companyx/control_file.scr "$@" >> /arsacif/companyx/arslog.out ;;
*) ;;
esac

exit 0

```

Tips: For more information about the codes for each message type that is logged in the system log, see Chapter 2, “Common Server Messages”, in *IBM Content Manager OnDemand - Messages and Codes*, SC27-1379. For example, message number 87 is listed as ARS0087I.

11.4.2 System log exit for z/OS

Content Manager OnDemand can be configured to record information, warning, and error messages. You can set up Content Manager OnDemand to record these messages by using the system log exit that is named the **ARSLOG** installation exit. The implementation of the system log exit on z/OS differs from the implementation on Multiplatforms. Like other z/OS exits, it uses the MVS Dynamic Exit Facility.

You configure the system log exit with the Administrator Client in the System Parameters window (see Figure 11-2 on page 254).

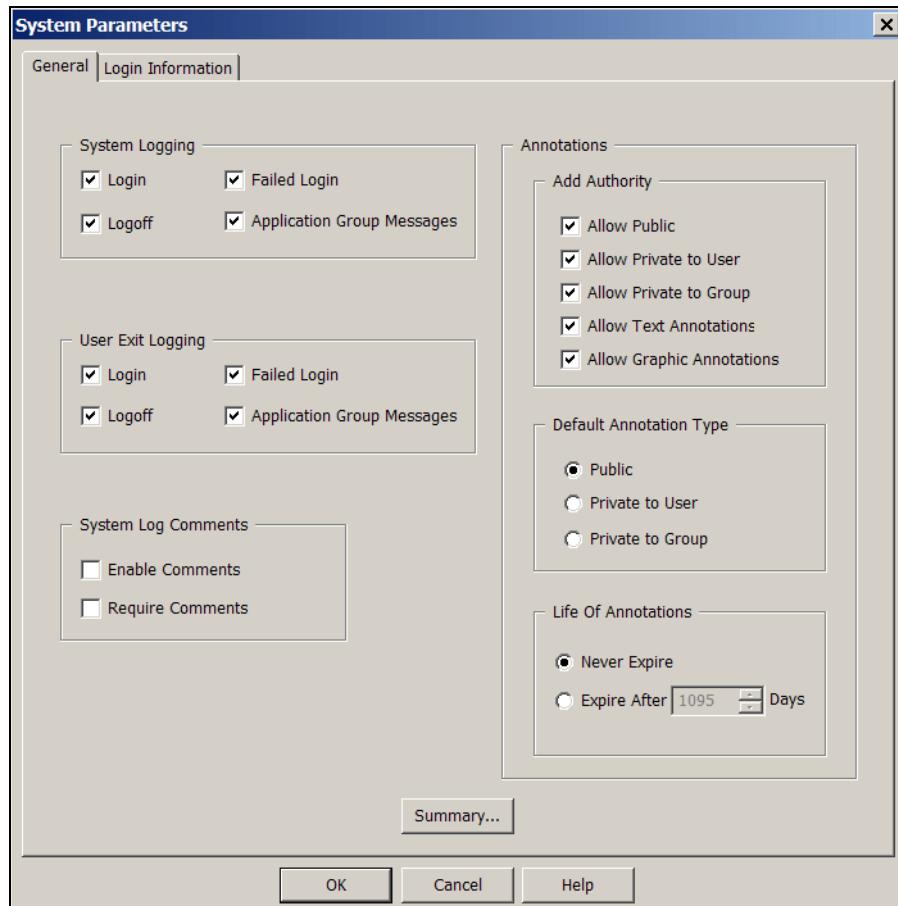


Figure 11-2 System Parameters window with User Exit Logging selected

Select the options for the system logging and set up the exit. The sample in Example 11-7 routes the messages to the system log with the write to operator (WTO) macro.

Example 11-7 System log exit setup sample

```

ARSLOG    title 'Issue a message to syslog'          00010000
***** START OF MODULE SPECIFICATIONS *****
*
*                                            * 00030000
*
*                                            * 00040000
*      ==> OD/390                      - 5655-H39 <=  * 00050007
*
*                                            * 00060000
*      Module Name:      ARSLOG           * 00070000
*
*      Descriptive Name: Issue a message to syslog * 00080000
*
*      Status:            Version ? Release ?   * 00090000
*
*      Function:          This routine issues a message to the SYSLOG * 00100000
*
*      Copyright:         5655-H39 (C) Copyright IBM Corp. 2013 * 00110000
*                          Licensed Materials-Property of IBM           * 00120000
*                          See Copyright instructions.                  * 00130000
*
*      Notes:              * 00140000
*
*      Restrictions:     None                 * 00150007
*
*                                            * 00160000
*                                            * 00170000
*                                            * 00180000
*                                            * 00190000
*                                            * 00200000
*                                            * 00210000
*                                            * 00220000

```

```

*      Register                                     * 00230000
*          Convention: R1 points to the Parameter list   * 00240000
*                                         R12 base register   00250000
*
*      Patch Label: PSPACE                         * 00260000
*
*      Input:             Parameter list pointed to by Register 1   * 00270000
*                           Parameter list contains addresses of:   * 00280000
*                                         - message length   * 00290000
*                                         - message text    * 00300000
*
*      Output:            None                      * 00310000
*                                         * 00320000
*                                         * 00330000
*
*      Return codes:                                * 00340000
*                                         * 00350000
*                                         * 00360000
*                                         * 00370000
*                                         * 00380000
*                                         * 00390000
*
*      NORMAL:           R15 = return code from WTO   * 00400000
*
*      Exits:            Return to caller via BR 14   * 00410000
*
*      External References:                         * 00420000
*                                         * 00430000
*
*      Change Activity: See below                  * 00440000
*                                         * 00450000
*
*      Ver Rel Mod     Date     Description of Change   * 00460000
*
*      0? 0? 00  04/05/00  Release ?.?
*
***** END OF MODULE SPECIFICATIONS *****
ARSLOG csect                               00500000
ARSLOG rmode any                            00510000
ARSLOG amode 31                             00520000
ARSLOG           using *,r15                00530000
ARSLOG           b          pastcopy          00540000
ARSLOG           dc         C'ARSLOG &sysdate' 00550000
ARSLOG           dc         C'5622-662 (C) COPYRIGHT IBM CORP. 2013' 00560000
ARSLOG           dc         C'ALL RIGHTS RESERVED' 00570000
ARSLOG           dc         C'LICENSED MATERIALS-PROPERTY OF IBM' 00580000
ARSLOG           ds         0h                 00590000
ARSLOG           stm        14,12,12(r13)    00600000
ARSLOG           lr         r12,r15          00610001
ARSLOG           lr         r2,r1             00620000
ARSLOG           using plist,r2            00630000
ARSLOG           drop       r15              00640000
ARSLOG           using ARSLOG,r12          00650000
ARSLOG           storage OBTAIN,length=work1,loc=ANY,cond=YES 00660000
ARSLOG           ltr        r15,r15          00670000
ARSLOG           jnz        bagit             00680000
ARSLOG           st         r13,4(,r1)        00690000
ARSLOG           st         r1,8(,r13)       00700000
ARSLOG           lr         r13,r1           00710000
ARSLOG           using workarea,r13        00720000
ARSLOG           *
ARSLOG           * Determine the message length 00730000
ARSLOG           *
ARSLOG           slr        r1,r1             Number of bytes 00740000
ARSLOG           l          r15,msgtxta      get starting address 00750005
ARSLOG           get       starting address 00760000
nulloop  ds         0h                 00770005
nulloop  cli        0(r15),x'00'        Is it zero? 00780005
nulloop  je         nomore            Yes - quit 00790006
nulloop  la         r1,1(,r1)         Bump count 00800005
nulloop  *

```

	la	r15,1(,r15)	bump address	00830005
	j	nulloop	And try next	00840005
nomore	ds	0h		00850005
	lr	r3,r1	Save length of message	00860005
	mvc	msgtxt+2(3),=c'XXX'	Set the prefix	00870007
	la	r14, msgtxt+5	Start to place number	00880005
	l	r15, msgnum	Get start of message number	00890005
numloop	ds	0h		00900005
	cli	0(r15),x'00'	Null?	00910005
	je	nomove		00920005
	mvc	0(0,r14),0(15)	move it	00930005
	la	r14,1(,r14)	next destination	00940005
	la	r15,1(,r15)	next source	00950005
	j	numloop	go do next	00960005
nomove	ds	0h		00970005
	l	r15,sev	Get severity	00980005
	cli	0(r15),c'1'	Is it Alert	00990005
	jne	tryerror	No skip	01000005
	mvi	0(r14),c'E'	Set error severity	01010006
	j	donesev		01020005
tryerror	ds	0h		01030005
	cli	0(r15),c'2'	"Error" severity?	01040005
	jne	trywarn	No - skip	01050005
	mvi	0(r14),c'E'	Set error	01060005
	j	donesev		01070006
trywarn	ds	0h		01080005
	cli	0(r15),c'3'	Is it Warning	01090006
	jne	setinfo		01100005
	mvi	0(r14),C'W'	Set Warning	01110005
	j	donesev		01120005
setinfo	ds	0h		01130005
	mvi	0(r14),c'I'	Indicate info	01140005
donesev	ds	0h		01150005
	mvi	1(r14),c' '	Put in blank	01160005
	la	r14,2(,r14)	Skip	01170005
	c	r3,=f'60'	More than 60 chars	01190005
	jnh	singlwto	No - issue it	01200005
	lhi	r3,60	Only first 60 chars	01210005
				01220005
	* We only need to issue a single WTO			01230005
				01240005
singlwto	ds	0h		01250005
	la	r4, msgtxt+2	Get start of text	01260005
	lr	r15,r14	Get where we stopped	01270005
	sr	r15,r4	Get how much we've done	01280005
	ar	r15,r3	add length of text	01290005
	stcm	r15,b'0011',msgtxt	Set the length	01300005
	bctr	r3,0	subtract 1	01310005
	l	r15, msgtxta	Get source address	01320005
	ex	r3,mvcins	Move it	01330005
				01340000
	mvc	wtoe,wtol	init the execute form	01350007
	la	r3, msgtxt		01360005
	slr	r0,r0		01370000
	wto	text=(r3),mf=(E,wtoe)		01380005
	j	exit	exit	01390000
				01400000
				02250000
exit	ds	0h		02260000

```

    1r    r1,r13                                02270000
    1    r2,4(r13)                             02280002
    storage RELEASE,length=work1,addr=(r1)      02290003
    1r    r13,r2                               02300002
    drop  r13                                 02310000
                                              02320000
bagit   ds    0h                                02330000
    1m    14,12,12(r13)                         02340001
    br    r14                                 02350000
psize   equ   ((*-ARSLOG+99)/100)*5          02360000
        dc    C'PATCH AREA - ARSLOG &sysdate'
pspace  dc    25s(*)                           02380000
        org   pspace                            02390000
        dc    ((psize+1)/2)s(*)                 02400000
                                              02410000
mvcins  mvc   0(0,r14),0(r15)                02420000
                                              02430000
                                              02450000
wto1    wto   text=,
        desc=(6),
        mcsflag=(BUSYEXIT),
        routcde=(11),
        mf=L                                +02460000
wto1l   equ   *-wto1                          02470000
        1torg                            +02480000
                                              +02490000
                                              02500000
wto1l   equ   *-wto1                          02510000
        1torg                            02520005
                                              02530005
workarea dsect                                02870000
rsa     ds    18f                                02880000
wtoe   ds    cl(wto1l)                          02890006
msgtxt ds    cl(72)                            02900005
work1  equ   *-workarea                        02910000
                                              02920000
plist   dsect                                02930000
instance ds   a                                02940005
tstamp  ds   a                                02950005
logrec  ds   a                                02960005
userid  ds   a                                02970005
acct    ds   a                                02980005
sev     ds   a                                02990005
msgnum  ds   a                                03000005
msgxta ds   a                                03010005
                                              03020005
yregs  ,
iezwpl
end   ,
                                              03030007
                                              03040005
                                              03050005

```

When the exit routine is assembled and link-edited to a library, it must be associated with the exit in one of two ways:

- ▶ Use the exit statement in the PROGXX parmlib member. For more information about the PROGXX parmlib member, see *z/OS MVS Initialization and Tuning Reference*, SA22-7592.
- ▶ Use the **SETPROG EXIT** operator command. For more information about the **SETPROG EXIT** command, see *z/OS MVS System Commands*, SA22-7627.

To activate the exit routine, run the following command:

```
SETPROG EXIT,ADD,EXITNAME=ARSLOG,MODENAME=ARSLOG, DSN=TEAM5.LOADLIB
```

The exit was link-edited to a normal library that is not AFP-authorized.

11.4.3 Print exit for Multiplatforms

A Content Manager OnDemand printer is an interface between the user and a print device that is controlled by a server. Multiple methods are available to print a document that is stored in Content Manager OnDemand:

- ▶ *Local printing*: This function is accomplished through a local area network (LAN)-attached personal computer printer.
- ▶ *Server printing*: This function is accomplished by submitting a print job to the print server queue, for example an IBM Infoprint Server print queue. Infoprint is an intelligent printer driver that provides AFP capabilities for Content Manager OnDemand servers.

A server print device can be physically connected to the library server or attached to another workstation in the network. Server print devices are managed by Infoprint.

Content Manager OnDemand provides a print exit for Multiplatforms that can be used only for documents that are printed through a *server printer*.

Two print exits are available for Multiplatforms, which are in the `bin` directory of the Content Manager OnDemand installation root for each platform:

- ▶ **arsprt**: Content Manager OnDemand User Exit Printing Facility
- ▶ **arsrdprt**: Content Manager OnDemand User Exit Printing Facility for Report Distribution

If you open either of the files in a text editor, you can see that they contain comments that provide a brief description of the exit and the order of the parameters that Content Manager OnDemand gives to this exit.

Example 11-8 shows an `arsprt` file, which updates application group indexes for a certain document type each time it is sent to a server printer. This example is from an actual customer where the requirement was for Content Manager OnDemand to keep a record of when a document is reprinted. This file is created by using the print exit to update the indexes of a document to show the last time that the document was reprinted and a counter is incremented to log the number of times the document was reprinted. Comments are inserted into the sample script in Example 11-8 that explain each part of the script. The customer name and the IP addresses are either altered or removed.

Example 11-8 Sample arsprt print exit file

```
#!/bin/ksh
#
#      arsprt - OnDemand User Exit Printing Facility
#
# 5622-662 (C) COPYRIGHT IBM CORPORATION 2013
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# This program sample is provided on an as-is basis.
# The license of the OnDemand product is free to copy, revise,
# modify, and make changes to this sample program
# as see fit.
#
# Function added to update a document each
# time a reprint is done. Index 'reprint' is updated with a 'I'
# and index 'log' is updated with a date and a counter of 001 (if the
# document has already been reprinted, the counter is added up by 001.
```

```

set -a
set -u
set -m
#set -x

#####
# 3 stmt's added #
# for debugging #
#####
#RANDOM=$$
#set -x
#exec 2> /usr/lpp/ars/bin/debug1.log.$RANDOM

RM=/bin/rm
SED=/bin/sed

OS=$(uname)

if [[ ${OS} = AIX ]] ; then
    BASE_DIR=/usr/lpp/ars/bin
elif [[ (${OS} = HP-UX) || (${OS} = SunOS) ]] ; then
    BASE_DIR=/opt/ondemand/bin
    ARSPRT_HOSTNAME=
else
    print "Cannot determine operating system"
    exit 1
fi

#
# $1 - Printer Queue Name
# $2 - Copies
# $3 - Userid
# $4 - Application Group Name
# $5 - Application Name
# $6 - Application Print Options
# $7 - Filename to Print

#
# NOTE: It is up to this script to make sure the file is deleted.
#       example( -r option on /bin/enq )
#
FILE=$7
OPTS_FILE=${FILE}.opts
NOTES_FILE=${FILE}.notes
if [[ -f ${OPTS_FILE} ]] ; then
    DEL=1
    PRT_OPTIONS="-o PASSTHRU=fax_file-${FILE}-"
    #
    # Since I am faxing, make sure messages are not produced.
    # If debugging is needed, then this parameter should be blank.
    #
    #EXTRA_OPTIONS="-o MSGCOUNT=0"
    EXTRA_OPTIONS="-o MSGCOUNT=0"
else
    DEL=0
    PRT_OPTIONS=
    EXTRA_OPTIONS=
fi

```

```

TITLE=$(print "$3 $4 $5" | ${SED} 's/-/ /g')

if [[ ${OS} = AIX ]] ; then
    /bin/enq -r -P "$1" -N $2 -T "${TITLE}" $6 ${EXTRA_OPTIONS} ${PRT_OPTIONS} ${FILE}
else
    ${BASE_DIR}/lprafp -p "$1" -s "${ARSPRT_HOSTNAME}" -o "COPIES=${2}" -o
"JOBNAME=${TITLE}" -o "TITLE=${TITLE}" $6 ${EXTRA_OPTIONS} ${PRT_OPTIONS} ${FILE}
fi

RC=$?

if [[ ${RC} = 0 ]] ; then
    if [[ ${OS} != AIX ]] ; then
        ${RM} -f ${FILE}

    else
        #####
        # Test if filename ends up with .0 #
        # If not, skip around code to update#
        # index. This prevents update of #
        # same index several times as only #
        # one .cntl file is created #
        # when server print is made for    #
        # multiple documents and this      #
        # script is called one time for    #
        # each doc to print.               #
        #####
        ext=$7
        ext=${ext##*.}
        if [[ ${ext} = 0 ]] ; then
            #####
            # Compute .cntl filename from      #
            # supplied parameter $7           #
            ######
            fil=$7
            mine=${fil%.*}.cntl
            #####
            # Double check if .cntl file exist #
            ######
            if test ! -f $mine
                then echo "File $mine not found"
                exit 1
            fi

            #####
            # Set static variables             #
            ######
            host=9.99.99.99
            nohit=no

            applgrp1=ICALog
            folder1=ICALog

            applgrp2=applg2
            folder2=folder2

            applgrp3=applg3
            folder3=folder3
            #####

```

```

# Read info from .cntl file      #
#####
cat $mine |grep -v APPLICATION|while read a1 a2 a3 a4 a5 a6 a7 a8 a9
do
#####
# Get the application group name  #
#####
applgrp=${a2}
applgrp=${applgrp##*=}
#####
# Set the folder name depending on #
# what the application group name  #
# is                                #
#####
if [[ ${applgrp} = ${applgrp1} ]]
then
    folder=$folder1
else
    if [[ ${applgrp} = ${applgrp2} ]]
    then
        folder=$folder2
    else
        if [[ ${applgrp} = ${applgrp3} ]]
        then
            folder=$folder3
        #####
        # Not an application group we are  #
        # looking for. Set nohits=yes to    #
        # skip to remove the .cntl file    #
        #####
        else
            nohit=yes
        fi
    fi
fi
#####
# If nohit=no, get Account-number and#
# log info                         #
#####
if [[ ${nohit} = no ]]
then
#####
# Get Account Number #
#####
account-number=${a4}
account-number=${account-number##*=}
#####
# Get log info. If first time,      #
# then set count=001 and current   #
# date                             #
#####
log=${a8}
log=${log##*=}
if [[ $log = "" ]]
then
    log=001
#####
# If not first time for reprint,   #
# then add up old count by 1      #
#####

```

```

        else
            let log=$log+001
            typeset -Z3 log
        fi
#####
# Set date after log count      #
#####
datum=`date +%Y-%m-%d`#
blank=" "
#####
# Update this document with count  #
# of reprints and current date   #
#####
arsdoc update -h $host -g $applgrp -f $folder -n log="$log$blank$datum" -n
reprint=I -u admin -p ondemand -i "where account-number='\$account-number'" -v
        fi
#####
# Done, remove the .cntl file    #
#####
done
rm $mine
fi

fi
else
(
    if [[ ${OS} = AIX ]] ; then
        echo /bin/enq -r -P "$1" -N $2 -T "${TITLE}" $6 ${EXTRA_OPTIONS} ${PRT_OPTIONS}
${FILE}
        else
            echo ${BASE_DIR}/lprafp -p "$1" -s "${ARSPRT_HOSTNAME}" -o "COPIES=${2}" -o
"JOBNAME=${TITLE}" -o "TITLE=${TITLE}" $6 ${EXTRA_OPTIONS} ${PRT_OPTIONS} ${FILE}
        fi

        echo "$(date)-->OnDemand Failed Print File >${FILE}< to Queue >$1<""
    ) >/dev/console
    exit ${RC}
fi

#
# If there is an options file, wait until the file has been
# printed before removing it.
#
if [[ ${DEL} != 0 ]] ; then
    while(( 1 ))
    do
        if [[ -f "${FILE}" ]] ; then
            sleep 30
        else
            ${RM} -f ${OPTS_FILE} ${NOTES_FILE}
            break
        fi
    done
fi
exit 0

```

11.5 Customized functions (Multiplatforms and z/OS only)

The user exits provide customized ways to perform tasks in Content Manager OnDemand. You can use a user exit to customize logins, retrieve data from external locations, or send a notification when a document is loaded. Programming of the user exits is an IBM Lab Services offering; for more information, contact IBM Lab Services.

You can also use the sample exit source code to write your own exits. In this section, we describe each sample exit that is provided in the *standard* Content Manager OnDemand installation.

The sample source code for the Content Manager OnDemand user exits is provided for all of the platforms. They are placed in the directories or libraries of Content Manager OnDemand that are listed in Table 11-1. These sample user exit modules provide a skeleton for you to program the exits.

Table 11-1 Exits and their initial locations

Module	Location
arsupload arsuprep arsuupd arsutbl	<p>Content Manager OnDemand V8.5:</p> <ul style="list-style-type: none">▶ Windows: C:\Program Files\IBM\OnDemand for Windows\bin\exits▶ AIX: /usr/lpp/ars/bin/exits▶ Solaris, HPUX, and Linux: /opt/ondemand/bin/exits▶ z/OS: /usr/lpp/ars/exits or ARS.V8R5M0.SARSINST <p>Content Manager OnDemand V9.0:</p> <ul style="list-style-type: none">▶ Windows: C:\Program Files\IBM\OnDemand for Windows\V9.0\bin\exits▶ AIX, Solaris, and HPUX: /opt/IBM/ondemand/V9.0/bin/exits▶ Linux and Linux on System z: /opt/ibm/ondemand/V9.0/bin▶ z/OS: /usr/lpp/ars/V9R0M0/exits or ARS.V9R0M0.SARSINST <p>Content Manager OnDemand V9.5:</p> <ul style="list-style-type: none">▶ Windows: C:\Program Files\IBM\OnDemand for Windows\V9.5\bin\exits▶ AIX, Solaris, and HPUX: /opt/IBM/ondemand/V9.5/bin/exits▶ Linux and Linux on System z: /opt/ibm/ondemand/V9.5/bin▶ z/OS: /usr/lpp/ars/V9R5M0/exits or ARS.V9R5M0.SARSINST

The header file provides information about how to turn on the user exits. If the information is not specified in the header file, place the compiled user exit program into the bin/exits directory of the Content Manager OnDemand installation root.

The source code must be compiled before you use it. For UNIX platforms, you can compile the source code by using the sample makefile that is provided. The makefile is in the same exits directory as the sample exits source code.

Table 11-2 provides the functions and usage of the user exit modules.

Table 11-2 User exits module

Module	Function	Usage
arsupload	LOADEXIT	To obtain load information for notification
arsuprep	PREPEXIT	To preprocess document data before document retrieval
arsuupd	ARSUUPDT	To alter parameters when document data is captured by ARSLOAD
arsutbl	TBLSPCRT	To customize the creation of table spaces, tables, and indexes

11.5.1 User exit header file (arccsxit.h)

Before you write the user exit, it is important to study the header file arccsxit.h. This file is in the same exits directory as the sample user exit source code. This file contains the structure and function declarations for the customized Content Manager OnDemand user exits. Also, instructions are included to activate the user exit after it is compiled.

The first part of the header file is a declaration of all of the structures and variables that are used. Example 11-9 shows several of the common structures that are used in the functions declarations.

Example 11-9 Common structure that is defined in the arccsxit.h header file

```
/***************************************************************************/  
/* COMMON STRUCTURES */  
/***************************************************************************/  
  
#define ARCCSXIT_MAX_SRVR_MESSAGE_SIZE 1024  
  
#if defined(AIX) || defined(HPUX) || defined(OS390) || defined(CICS)  
    #define ARCCSXIT_PATH_MAX 1023  
#elif defined(LINUX)  
    #define ARCCSXIT_PATH_MAX 4096  
#elif defined(SUNOS) || defined(_OS400_)  
    #define ARCCSXIT_PATH_MAX 1024  
#elif defined(WIN32)  
    #define ARCCSXIT_PATH_MAX 260  
#endif  
  
typedef struct _ArcCSxitApplGroup  
{  
    char    *name;  
    ArcI32  agid;  
    char    *agid_name;  
} ArcCSxitApplGroup;  
  
typedef struct _ArcCSxitApplGroupU  
{  
    ArcChar *name;  
    ArcI32  agid;  
    ArcChar *agid_name;  
} ArcCSxitApplGroupU;  
  
typedef ArcU8 ArcCSxitDocType;  
#define ARCCSXIT_DOC_TYPE_AFP      (ArcCSxitDocType) 0x41  
#define ARCCSXIT_DOC_TYPE_BMP     (ArcCSxitDocType) 0x42  
#define ARCCSXIT_DOC_TYPE_EMAIL   (ArcCSxitDocType) 0x45  
#define ARCCSXIT_DOC_TYPE_GIF     (ArcCSxitDocType) 0x47  
#define ARCCSXIT_DOC_TYPE_JFIF    (ArcCSxitDocType) 0x4A  
#define ARCCSXIT_DOC_TYPE_DJDE    (ArcCSxitDocType) 0x4B  
#define ARCCSXIT_DOC_TYPE_LINE    (ArcCSxitDocType) 0x4C  
#define ARCCSXIT_DOC_TYPE_META    (ArcCSxitDocType) 0x4D  
#define ARCCSXIT_DOC_TYPE_NONE    (ArcCSxitDocType) 0x4E  
#define ARCCSXIT_DOC_TYPE_ODDOC   (ArcCSxitDocType) 0x4F  
#define ARCCSXIT_DOC_TYPE_PCX     (ArcCSxitDocType) 0x50  
#define ARCCSXIT_DOC_TYPE_PDF     (ArcCSxitDocType) 0x52  
#define ARCCSXIT_DOC_TYPE_PNG     (ArcCSxitDocType) 0x51  
#define ARCCSXIT_DOC_TYPE_SCS     (ArcCSxitDocType) 0x53  
#define ARCCSXIT_DOC_TYPE_SCS_EXT (ArcCSxitDocType) 0x58  
#define ARCCSXIT_DOC_TYPE_TIFF    (ArcCSxitDocType) 0x54  
#define ARCCSXIT_DOC_TYPE_USRDEF  (ArcCSxitDocType) 0x55
```

```

typedef ArcU8 ArcCSxitDocFormat;
#define ARCCSXIT_DOC_FORMAT_FIXED    (ArcCSxitDocFormat) 0x00
#define ARCCSXIT_DOC_FORMAT_VARIABLE (ArcCSxitDocFormat) 0x01
#define ARCCSXIT_DOC_FORMAT_STREAM   (ArcCSxitDocFormat) 0x02

typedef ArcU8 ArcCSxitCarCtl;
#define ARCCSXIT_CC_ANSI    (ArcCSxitCarCtl) 'A'
#define ARCCSXIT_CC_MACHINE (ArcCSxitCarCtl) 'M'
#define ARCCSXIT_CC_NONE    (ArcCSxitCarCtl) 'N'

typedef ArcU8 ArcCSxitPrMode;
#define ARCCSXIT_PRMODE_NONE   (ArcCSxitPrMode) 'N'
#define ARCCSXIT_PRMODE_SOSI1  (ArcCSxitPrMode) '1'
#define ARCCSXIT_PRMODE_SOSI2  (ArcCSxitPrMode) '2'
#define ARCCSXIT_PRMODE_SOSI3  (ArcCSxitPrMode) '3'

typedef struct _ArcCSxitAppl
{
    char          *name;
    ArcI32        aid;
    ArcCSxitDocType doc_type;
    ArcCSxitDocFormat doc_fmt;      /* Document Format for Linedata */
    union
    {
        ArcI32 fixed;           /* Fixed - Record Length */
        char   stream[17];      /* Stream - Character Delimiters */
    } u;
    ArcU8         trc_present; /* 0 = no, 1 = yes */
    ArcI32        line_count; /* Lines per page for line data */
    ArcI32        code_page;  /* Code Page for line data */
    ArcCSxitCarCtl cc_type;   /* CC type for line data */
    ArcCSxitPrMode prmode;    /* PRMode for line data */
} ArcCSxitAppl;

typedef struct _ArcCSxitApplU
{
    ArcChar        *name;
    ArcI32        aid;
    ArcCSxitDocType doc_type;
    ArcCSxitDocFormat doc_fmt;      /* Document Format for Linedata */
    union
    {
        ArcI32 fixed;           /* Fixed - Record Length */
        ArcChar stream[17];      /* Stream - Character Delimiters */
    } u;
    ArcU8         trc_present; /* 0 = no, 1 = yes */
    ArcI32        line_count; /* Lines per page for line data */
    ArcI32        code_page;  /* Code Page for line data */
    ArcCSxitCarCtl cc_type;   /* CC type for line data */
    ArcCSxitPrMode prmode;    /* PRMode for line data */
} ArcCSxitApplU;

typedef ArcU8 ArcCSxitFieldType;
#define ARCCSXIT_FIELD_TYPE_BIGINT     (ArcCSxitFieldType) 0x42
#define ARCCSXIT_FIELD_TYPE_DATE       (ArcCSxitFieldType) 0x61
#define ARCCSXIT_FIELD_TYPE_DATETIME   (ArcCSxitFieldType) 0x62
#define ARCCSXIT_FIELD_TYPE_DECFLOAT16 (ArcCSxitFieldType) 0x38
#define ARCCSXIT_FIELD_TYPE_DECFLOAT34 (ArcCSxitFieldType) 0x39
#define ARCCSXIT_FIELD_TYPE_DECIMAL   (ArcCSxitFieldType) 0x44

```

```

#define ARCCSXIT_FIELD_TYPE_INTEGER      (ArcCSXitFieldType) 0x49
#define ARCCSXIT_FIELD_TYPE_SMALLINT    (ArcCSXitFieldType) 0x4E
#define ARCCSXIT_FIELD_TYPE_STRING      (ArcCSXitFieldType) 0x53

typedef ArcU8 ArcCSXitFieldTypeQual;
#define ARCCSXIT_FIELD_TYPE_QUAL_BASE    (ArcCSXitFieldTypeQual) 0x42
#define ARCCSXIT_FIELD_TYPE_QUAL_DATETIME (ArcCSXitFieldTypeQual) 0x43
#define ARCCSXIT_FIELD_TYPE_QUAL_DATE     (ArcCSXitFieldTypeQual) 0x44
#define ARCCSXIT_FIELD_TYPE_QUAL_TIME     (ArcCSXitFieldTypeQual) 0x54
#define ARCCSXIT_FIELD_TYPE_QUAL_TZ_DATETIME (ArcCSXitFieldTypeQual) 0x5A

typedef struct _ArcCSXitField
{
    char                  *db_name;
    ArcCSXitFieldType      type;
    ArcCSXitFieldTypeQual   qual;
    union
    {
        ArcI16          n;
        ArcI32          i;
        ArcI64          b;
        double          d;
        char            *str;
        ArcDateTime     dt;
        ArcDecimal64    d64;
        ArcDecimal128   d128;
    } u;
} ArcCSXitField;

typedef struct _ArcCSXitFieldU
{
    ArcChar                *db_name;
    ArcCSXitFieldType      type;
    ArcCSXitFieldTypeQual   qual;
    union
    {
        ArcI16          n;
        ArcI32          i;
        ArcI64          b;
        double          d;
        ArcChar         *str;
        ArcDateTime     dt;
        ArcDecimal64    d64;
        ArcDecimal128   d128;
    } u;
} ArcCSXitFieldU;

typedef struct _ArcCSXitDocFields
{
    ArcI32          flds_num;
    ArcCSXitField  *flds;
} ArcCSXitDocFields;

#define ARCCSXIT_DOCNAME_SIZE 11

typedef struct _ArcCSXitDocHandle
{
    char  name[ARCCSXIT_DOCNAME_SIZE + 1];
    ArcU32 doc_off;
    ArcU32 doc_len;
}

```

```

    ArcU32 comp_off;
    ArcU32 comp_len;
} ArcCSxitDocHandle;

typedef struct _ArcCSxitDoc
{
    ArcCSxitDocFields doc_flds;
    ArcCSxitDocHandle doc_hdl;
} ArcCSxitDoc;

```

From the previous example, the *ArcCSxitApplGroup* structure consists of the application group name, the application group identifier (agid), and the AGID name (agid_name). This information is important because it indicates the input to the functions. Structures that are specific to a function are also included in the header file.

In the following sections, we examine each exit and describe its usage.

11.5.2 Load exit

The load exit is used to send a notification after a document is loaded. The header file in Example 11-10 shows the information that can be incorporated into the notification message.

Example 11-10 Header file of the load exit

```

/******************************************************************************/
/* LOADEXIT - Load Exit
*/
/* To activate the load exit, the arsupload dll must exist in the
/* OnDemand exits installation directory.
*/
/* INPUT: load
*/
/* OUTPUT:
/*      None
*/
/* RETURN_CODE:
/*      0          -> Successful
/*      Otherwise -> Failed
*/
/******************************************************************************/
typedef struct _ArsCSxitLoadExit
{
    char            *hostname; /* OnDemand Library Server Hostname */
    char            *load_id;  /* Load Id */
    ArcU32         deprecated; /* was bytes. Use report_bytes */
    ArcU32         res_bytes; /* Number of resource bytes stored */
    ArcCSxitApplGroup *appl_grp; /* Application Group Info */
    ArcCSxitAppl   *appl;    /* Application Info */
    char            *file;    /* File containing all rows */
    char            *user_def; /* User Specified string to load */
    ArcCSxitField  *reference; /* Reference column defined for ODF */
    char            *file_l;   /* File containing rows in non-UTF8 */
    ArcU32         cp;       /* codepage file_l is in */
    void           **hdl;    /* pointer to anchor for arsupload */
    char            ColDelim; /* Character used to delimit columns*/
    ArcI64         report_bytes; /* Number of bytes in report */
    char            *instance; /* OD Instance name */
} ArsCSxitLoadExit;

```

```
ArcI32
ARSCSXIT_EXPORT
ARSCSXIT_API
LOADEXIT( ArsCSxitLoadExit *load );
```

You can use the sample exits program to insert the action that you prefer. The input to the program is in the structure *ArsCSxitLoadExit*. This structure contains the load information, such as the load identifier and the application group name. Based on the load information, you decide whether to send a notification, to whom to send the notification, and the type of information you want to provide when loading is successful.

Activating the load exit

To activate the exits, place the compiled exit program **arsuload** in the bin/exits directory of the Content Manager OnDemand installation root.

Client retrieval preview exit

Use the client retrieval preview user exit to modify document data before the data is presented to a client. This exit is called during the retrieval of a document.

You can use the client retrieval preview exit to add, remove, or reformat data before the document is presented to the client, for example:

- ▶ You can remove pages from the document, such as banner pages, title pages, or all pages except the summary page.
- ▶ You can remove specific words, columns of data, or other information from the document. That is, you can omit ("white out") sensitive information, such as salaries, social security numbers, and birth dates.
- ▶ You can add information to the document, for example, a summary page, data analysis information, and Confidential or Copy statements.
- ▶ You can reformat data that is contained in the document. For example, you can reorder the columns of data.

The client retrieval preview exit point might be enabled for specific applications. However, to enable the client retrieval preview exit for a specific application, ensure that the Use Preview Exit option is selected on the Miscellaneous Options page of the application.

The input to the exit program is captured when the user tries to retrieve the document. Based on the input, such as application group name and the indexes, you can then use your program to create an output file with the name from pOutFileName.

Example 11-11 shows the header file of the client retrieval preview exit.

Example 11-11 Header file of client retrieval preview exit

```
*****
/* PREPEXIT - Client Retrieval Preview Exit */
/*
/*  This exit is used to modify the contents of a document prior */
/* retrieving the document */
/*
/* INPUT:
/*      pInFileName
/*      pOutFileName
/*      pUserParms
/*      pApplGrp
/*      pApp1
*/
```

```

/*
/*          pDoc
/*
/* OUTPUT:
/*
/* RETURN_CODE:
/*      0          -> Successful
/*      Otherwise -> Failed
/*
/* ****
typedef struct _ArsCSxitPrepExit
{
    char          *pUserId;      /* Logged on userid */
    char          *pInFileName; /* File name for document data */
    char          OutFileName[ARCCSXIT_PATH_MAX + 1];
                                /* File name for modified data */
    char          *pUserParms;   /* User defined parms from appl */
    ArcCSxitApplGroup *pApplGrp; /* Appl Grp info */
    ArcCSxitAppl    *pAppl;     /* Application info */
    ArcCSxitDoc     *pDoc;      /* Doc handle, field info */
    char          *instance;    /* OD Instance name */
} ArsCSxitPrepExit;

ArcI32
ARSCSXIT_EXPORT
ARSCSXIT_API
PREPEXIT( ArsCSxitPrepExit *prep );

```

For example, you can arrange it so that when a user retrieves a document from a particular application group, you can check the name of the account number (the indexes from the Doc handle) and place a watermark for that document. When the document is retrieved by the user, the user sees the document with the watermark.

Activating the client retrieval preview exit

To activate the client retrieval preview exit, select the **Use Preview Exit** option on the Miscellaneous Options page of an application and place the exit in the bin/exits directory of the Content Manager OnDemand installation root. When the option is selected, the user-written program is called any time that a request is made to retrieve a document.

Any information that is specified in the Parameters field is passed to the user-written program. Place the **arsuprep** program in the bin/exits directory.

The client retrieval preview user exit can be enabled for all data types, except for None.

For more information, see the *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.

11.5.3 Report specifications archive definition exit

Use the Content Manager OnDemand report specifications archive definition exit to change several of the parameters that are used by Content Manager OnDemand when document data is loaded by the **ARSLOAD** program. **ARSUUPDT** is a dynamic link library (DLL) module that is written in the C programming language.

The first call modifies the names process for parameters, such as application group, application, object server, storage node, DB field date format, and DB field name. The second call modifies the indexing parameters and input file parameters. Example 11-12 shows the header file of the report specifications archive definition exit.

Example 11-12 Header file of the report specifications archive definition exit

```
*****  
/* UPDTEXIT - Report Definition Update Exit */  
/*  
/* This exit is for specialized applications and is not normally */  
/* used. */  
/*  
/* INPUT:  
/*      pFileName */  
/*      Function */  
/*      ApplGrpName */  
/*      ApplName */  
/*      ObjServer */  
/*      StorageNode */  
/*      pJES */  
/*      IndexerParms */  
/*      CCType */  
/*      LRECL */  
/*      RECFM */  
/*      Delim */  
/*      instance */  
/*  
/* OUTPUT:  
/*      ApplGrpName */  
/*      ApplName */  
/*      ObjServer */  
/*      StorageNode */  
/*      IndexerParms */  
/*      CCType */  
/*      LRECL */  
/*      RECFM */  
/*      UpdateAppl */  
/*      Delim */  
/*      DbFieldName */  
/*      DbFieldDateFormat */  
/*  
/* RETURN_CODE:  
/*      0          -> Successful */  
/*      Otherwise -> Failed */  
/*  
*****  
  
#if defined(OS390)  
typedef struct _ArsCSxitUpdtExit_JES  
{  
    void *JES_SSS2p; /* pointer to SSS2 (SAPI SSOB ext) */  
    char JES_DD8"; /* DD name allocated to spool file */  
} ArsCSxitUpdtExit_JES;  
#endif  
  
typedef struct _ArsCSxitUpdtExit  
{  
    char          *pFileName;  
    ArcI32        Function;
```

```

    char          ApplGrpNameÝARCCSXIT_MAX_NAME_SIZE + 1";
    char          ApplNameÝARCCSXIT_MAX_NAME_SIZE + 1";
    char          ObjServerÝARCCSXIT_MAX_SERVER_SIZE + 1";
    char          StorageNodeÝARCCSXIT_MAX_NAME_SIZE + 1";
    void          *pJES;
    char          IndexerParmsÝARCCSXIT_MAX_INDEXER_SIZE + 1";
    ArcCSxitCarCtl CCType;
    ArcI32        LRECL;
    ArcCSxitDocFormat RECFM;
    ArcI32        UpdateAppl;
    char          DelimÝARCCSXIT_MAX_DELIMITER_SIZE + 1";
    char          *instance;
    char          DbFieldNameÝARCCSXIT_MAX_DBCOL_NAME_SIZE + 1";
    char          DbFieldDateFormatÝARCCSXIT_MAX_DATEFMT_SIZE + 1";
} ArsCSxitUpdtExit;

typedef struct _ArsCSxitUpdtExitU
{
    ArcChar        *pFileName;
    ArcI32         Function;
    ArcChar        ApplGrpNameÝARCCSXIT_MAX_NAME_SIZE + 1";
    ArcChar        ApplNameÝARCCSXIT_MAX_NAME_SIZE + 1";
    ArcChar        ObjServerÝARCCSXIT_MAX_SERVER_SIZE + 1";
    ArcChar        StorageNodeÝARCCSXIT_MAX_NAME_SIZE + 1";
    void          *pJES;
    ArcChar        IndexerParmsÝARCCSXIT_MAX_INDEXER_SIZE + 1";
    ArcCSxitCarCtl CCType;
    ArcI32        LRECL;
    ArcCSxitDocFormat RECFM;
    ArcI32        UpdateAppl;
    ArcChar        DelimÝARCCSXIT_MAX_DELIMITER_SIZE + 1";
    ArcChar        *instance;
    ArcChar        DbFieldNameÝARCCSXIT_MAX_DBCOL_NAME_SIZE + 1";
    ArcChar        DbFieldDateFormatÝARCCSXIT_MAX_DATEFMT_SIZE + 1";
} ArsCSxitUpdtExitU;

ArcI32
ARSCSXIT_EXPORT
ARSCSXIT_API
UPDTEXIT( ArsCSxitUpdtExit *updt );

```

Activating the report specifications archive definition exit

The report specifications archive definition exit is implemented by a single DLL, **ARSUUPDT**. **ARSUUPDT** is a DLL module that is written in the C programming language. The samples that are shipped (**ARSUUPDT** and **ARSUUPDC**) initialize the **ARSUUPDA** structure and call the **ARSUUPDX** **ARS.RSADUPDT** exit driver.

The **ARSUUPDT** DLL invokes module **ARSUUPDX**. Module **ARSUUPDX** interfaces with the MVS Dynamic Exit Facility to perform the following actions:

- ▶ Define the logical exit point name: **ARS.RSADUPDT**
- ▶ Route control to a set of exit routines that are associated with MVS and process the results of their execution

Module **ARSUUPDZ** is implemented as dynamic exit routine that is associated with MVS. An exit routine is eligible for execution after it becomes associated with the logical exit point. The MVS Dynamic Exit Facility provides several methods for performing this association.

When the exit routine is assembled and link-edited to a library, it must be associated with the exit in one of two ways:

- ▶ Use the exit statement in the PROGXX parmlib member. For more information about the PROGXX parmlib member, see *z/OS MVS Initialization and Tuning Reference*, SA22-7592.
- ▶ Use the **SETPROG EXIT** operator command. For more information about the **SETPROG EXIT** command, see *z/OS MVS System Commands*, SA22-7627.

Use the following command to activate the exit routine and associate **ARSUUPDZ** with the logical exit point name. (The example assumes that **ARSUUPDZ** is in the link pack area (LPA) or a LNKLST dataset.)

```
SETPROG EXIT,ADD,EXITNAME=ARS.RSADUPDT,MODNAME=ARSUUPDZ
```

Enabling the report specifications archive definition exit

To enable the exit in Content Manager OnDemand, run the **ARSLLOAD** program with the **-E** parameter.

Note: The **-E** parameter must be specified in uppercase.

For more information about the report specifications archive definition exit routines, see Chapter 40, “Report specifications archive definition exit”, in the *Content Manager OnDemand for z/OS Configuration Guide*, SC19-3363.

11.5.4 Table space creation exit

With the Content Manager OnDemand table space creation exit, you can act when Content Manager OnDemand creates a table space, table, or index tables that are used to store application index data. The exit is not called for the Content Manager OnDemand system tables. The table space creation exit is used to modify the way Content Manager OnDemand creates table spaces, tables, or indexes. For table and index creation, you can alter the SQL that is used to create the table or index.

You can also use this exit to perform other actions during a table space creation. This exit is useful if you must change default parameters for the table space, the table, or the indexes. The changes affect only new creations. Example 11-13 shows the header file of the table space creation exit.

Example 11-13 Header file for the table space creation exit

```
/********************************************/  
/* TBLSPCRT - table space Create Exit */  
/* */  
/* To activate the table space creation exit, set the following */  
/* variable in the appropriate OnDemand instance ars.cfg file: */  
/* */  
/* ARS_DB_TABLESPACE_USEREXIT=<absolute_dll_path_name> */  
/* */  
/* INPUT: appl_grp */  
/* tblsp_name */  
/* table_name */  
/* idx_name */  
/* sql (allocated with 16384 bytes) */  
/* action */  
/* instance */  
/* */
```

```

/* OUTPUT: */ */
/*
 1) OnDemand will invoke the exit with action == 1 */
/* so that the exit can create the table space (tblsp_name) */
/* using (sql) */
/* *created -> 0 exit did not create the table space, */
/* OnDemand needs to create the table space */
/* using (sql), which can be left unchanged */
/* or modified by the exit */
/* *created -> 1 exit created the table space */
/*
 2) OnDemand will then invoke the exit with action == 2 */
/* so that the exit can create the table (table_name) */
/* inside of the table space (tblsp_name) using (sql) */
/* *created -> 0 exit did not create the table, */
/* OnDemand needs to create the table */
/* using (sql), which can be left unchanged */
/* or modified by the exit */
/* *created -> 1 exit created the table */
/*
 3) OnDemand will then invoke the exit with action == 3 */
/* so that the exit can create the table indexes (idx_name) */
/* inside of the table space (tblsp_name) for table */
/* (table_name) using (sql). This will be invoked based */
/* on the number of indexes to create for the appl_grp */
/* *created -> 0 exit did not create the index, */
/* OnDemand needs to create the index */
/* using (sql), which can be left unchanged */
/* or modified by the exit */
/* *created -> 1 exit created the index */
/*
 4) OnDemand will then invoke the exit with action == 4 */
/* so that the exit can perform any additional work */
/* *created -> Is not used */
/* sql -> If sql is not an empty string, OnDemand */
/* will issue (sql) to the database */
/*
 If ARS_DB_TABLESPACE_USEREXIT_EXTRA=1 is defined in */
/* ars.cfg, then the following actions will also be invoked */
/* when OnDemand needs to do further actions:
*/
/*
 5) OnDemand will invoke the exit with action == 5 */
/* so that the exit can drop the table space (tblsp_name) */
/* using (sql) */
/* *created -> 0 exit did not drop the table space, */
/* OnDemand needs to drop the table space */
/* using (sql), which can be left unchanged */
/* or modified by the exit */
/* *created -> 1 exit dropped the table space */
/*
 6) OnDemand will invoke the exit with action == 6 */
/* so that the exit can drop the table (table_name) */
/* using (sql) when OnDemand needs to drop a table */
/* *created -> 0 exit did not drop the table, */
/* OnDemand needs to drop the table */
/* using (sql), which can be left unchanged */
/* or modified by the exit */
/* *created -> 1 exit dropped the table */
/*
 7) OnDemand will invoke the exit with action == 7 */

```

```

/*
 *      so that the exit can drop the index (idx_name)      */
/*      using (sql)                                         */
/*      *created -> 0 exit did not drop the index,          */
/*              OnDemand needs to drop the index           */
/*      using (sql), which can be left unchanged          */
/*      or modified by the exit                           */
/*      *created -> 1 exit dropped the index             */
/*                                                       */
/*      8) OnDemand will invoke the exit with action == 8   */
/*      so that the exit can alter the table (table_name)  */
/*      using (sql)                                         */
/*      *created -> 0 exit did not alter the table,        */
/*              OnDemand needs to alter the table          */
/*      using (sql), which can be left unchanged          */
/*      or modified by the exit                           */
/*      *created -> 1 exit altered the table             */
/*                                                       */
/* RETURN_CODE:                                         */
/*      0          -> Successful                         */
/*      Otherwise -> Failed                            */
/*                                                       */
/*************************************************************************************************/
ArcI32
ARSCSXIT_EXPORT
ARSCSXIT_API
TBLSPCRT( ArcCSxitApplGroup *appl_grp,
           char *tblsp_name,
           char *table_name,
           char *idx_name,
           char *sql,
           ArcI32 action,
           ArcI32 *created,
           char *instance
 );

```

You can use SQL code to customize the following actions:

- ▶ Creating a table space
- ▶ Creating a table
- ▶ Creating an index
- ▶ Other additional action

If you do not customize the action, Content Manager OnDemand uses the defaults.

Example 11-14 shows a sample program flow.

Example 11-14 Sample program flow

```

Action 1
Is there a need to customize the creation of the table space?
  If yes
    create the tablespace
    return( created = 1)
  Else
    OnDemand create the tablespace
    return( created = 0)

```

```

Action 2
Is there a need to customize the creation of the table?
  If yes

```

```

        create the table (in the tablespace)
        return( created = 1 )
    Else
        OnDemand create the table
        return( created = 0 )

Action 3
Is there a need to customize the creation of the indexes?
If yes
    create the indexes
    return( created = 1 )
Else
    OnDemand create the indexes
    return( created = 0 )

Action 4
Final call, is there additional work, clean up or update on parameters?
If yes
    perform the additional action.
    return( created = not used )
Else
    OnDemand do nothing
    return( created = not used )

```

Activating the table space creation exit

The exit is turned on by setting the following parameter in the ARS.CFG file, which is in the config directory of the Content Manager OnDemand installation root.

The following statement must exist in the ARS.CFG file that is associated with the instance so that the ARSUTBL DLL can be invoked:

ARS_DB_TABLESPACE_USEREXIT=absolute path name

Where “absolute path name = ... /bin/exits/arsutbl”

For more information about the table space creation exit, see the *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC18-9232.

11.5.5 ARSYSPIN and sample APKACIF exit on z/OS

The JES Spool Capture facility **ARYSPIN** and the sample **APKACIF** exit are provided on z/OS. **ARYSPIN** provides a means to collect and consolidate the JES spool (SYSOUT) dataset into one or more files so they can be archived by Content Manager OnDemand. The facility runs as a started task in its own address space. A control statement file is used to provide **ARYSPIN** parameters. These parameters specify JES Spool file selection criteria (for example, the sysout class that is taken for Capture output) and other operational characteristics.

ARYSPIN creates an intermediate output file that contains one or more spool files from one or more jobs. The intermediate output file is indexed and stored in Content Manager OnDemand by using the **ARSLOAD** program. **ARYSPIN** invokes **ARSLOAD** when sufficient data is captured in the intermediate output file. **ARSLOAD** calls the indexer program (**APKACIF**) to extract the index values from the data and store them in an index file. **ARSLOAD** adds these index values to the database and stores the data object. If you want, you can use **ARYSPIN** exits to augment the data stream.

In particular, the **ARSYSPIN** Input Exit (UX03) and Separator Exit (UX06) provide substantially more information about the job that produced the spool file that is being processed than what is available at the time when **APKACIF** (or another indexer program) is driven by Content Manager OnDemand. In addition, the processing impact of driving **ARSYSPIN** exit routines is lower than the impact that is associated with the indexer exit routines, such as **ARSSPVIN**.

ARSSPVIN is a sample **APKACIF** input exit that is provided with **ARSYSPIN** to introduce additional index values into the data stream, by using a “trailer” record. Trailer records are inserted at the end of the JESMSGLG data. They reflect the highest severity condition (a step completion code, an ABEND code, or another type of problem, such as a JCL error) that is observed in messages that are contained within these spool files.

Special considerations for APKACIF exits that are written in COBOL

The provided sample exit is written as a COBOL main program. To prevent the IBM Language Environment® from creating and destroying the COBOL runtime environment each time that **ARSSPVIN** is called, a CEEUOPT CSECT must be assembled and link-edited with the COBOL object code.

Constructing a CEEUOPT CSECT is documented in *z/OS Language Environment Customization*, SA22-7564. A sample CEEUOPT CSECT is included in dataset CEE.SCEESAMP(CEEUOPT). You can use this sample as a model, but you must ensure that the following option is specified:

RTEREUS=(ON)

CEEUOPT CSECT must be assembled and link-edited with the COBOL object code. In addition, you must ensure that the resulting module is link-edited as NOT RE-ENTRANT and NOT REUSEABLE. This task is required for the local variables within the COBOL exit code to retain their values. This exit is invoked several times during an ACIF run. The sample source code is in the SARSINST library member ARSSPVIN. Example 11-15 shows a sample CEEUOPT CSECT.

Example 11-15 CEEUOPT CSECT

```
CEEUOPT CSECT ,
CEEUOPT AMODE ANY
CEEUOPT RMODE ANY
CEEXOPT
ABPERC=(NONE), +
ABTERMENC=(ABEND), +
AIXBLD=(OFF), +
ALL31=(ON), +
ANYHEAP=(16K, 8K, ANYWHERE, FREE) +
BELOWHEAP=(8K, 4K, FREE), +
CBLOPTS=(ON), +
CBLPSHPOP=(ON), +
CBLQDA=(OFF), +
CEEDUMP=(60,SYSOUT=*,FREE=END,SPIN=UNALLOC), +
CHECK=(ON), +
COUNTRY=(US), +
DEBUG=(OFF), +
DEPTHCONDLMT=(10), +
DYNDUMP=(*USERID,NODYNAMIC,TDUMP), +
ENVAR=('), +
ERRCOUNT=(0), +
ERRUNIT=(6), +
FILEHIST=(ON), +
FILETAG=(NOATOCVT,NOAUTOTAG), +
HEAP=(32K,32K,ANYWHERE,KEEP,8K,4K), +
```

```

HEAPCHK=(OFF,1,0,0,0), +
HEAPPOLLS=(OFF,8,10,32,10,128,10,156,10,1024,10,2048, +
10,0,10,0,10,0,10,0,10,0,10,0,10), +
INFOMSGFILTER=(OFF,,,), +
INQPCOPN=(ON), +
INTERRUPT=(OFF), +
LIBSTACK=(4K,4K,FREE), +
MSGFILE=(SYSOUT,FBA,121,0,NOENQ), +
MSGQ=(15), +
NATLANG=(ENU), +
NOAUTOTASK=,
NOTE=(ALL,*,PROMPT,INSPPREF), +
NOUSRHDLR(''), +
OCSTATUS=(ON), +
PC=(OFF), +
PLITASKCOUNT=(20), +
POSIX=(OFF), +
PROFILE=(OFF,''), +
PRTUNIT=(6), +
PUNUNIT=(7), +
RDRUNIT=(5), +
RECPAD=(OFF), +
RPTOPTS=(OFF), +
RPTSTG=(OFF), +
RTEREUS=(ON), + <====ATTENTION
SIMVRD=(OFF), +
STACK=(128K,128K,ANYWHERE,KEEP,512K,128K), +
STORAGE=(NONE,NONE,NONE,OK), +
TERMTHDACT=(TRACE,,96), +
THREADHEAP=(4K,4K,ANYWHERE,KEEP), +
THREADSTACK=(OFF,4K,4K,ANYWHERE,KEEP,128K,128K), +
TRACE=(OFF,4KDUMP,LE=0), +
TRAP=(ON,SPIE), +
UPSI=(00000000), +
VCTRSAVE=(OFF), +
XPLINK=(OFF), +
XUFLOW=(AUTO) +
END,

```

Activating the exit

To activate the exit, you must add the executable file to a loadlib in the Steplib (**ARSLOAD**) procedure. You must also supply the ACIF control statement INPEXIT = ARSSPVIN to the indexing parameters. You can perform this task when you add an application in the Indexer Information window.

Complete the following steps:

1. Open the Add an Application window and click the **Indexer Information** tab, as shown in Figure 11-3.

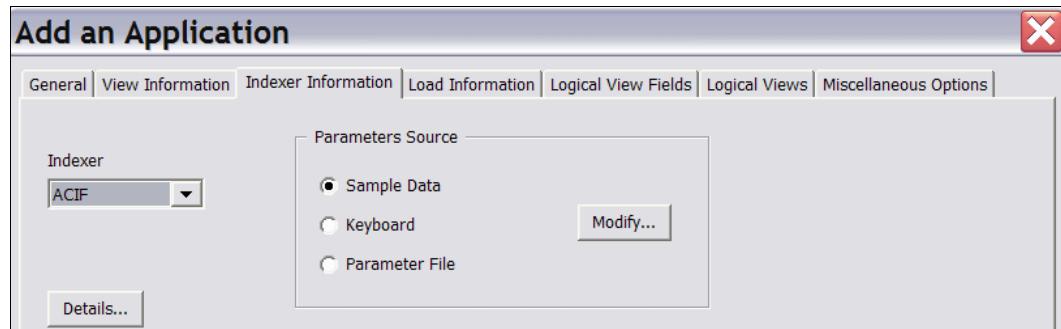


Figure 11-3 Indexer Information tab

2. Click **Modify**.
3. Click the **Exit Information** tab, as shown in Figure 11-4.

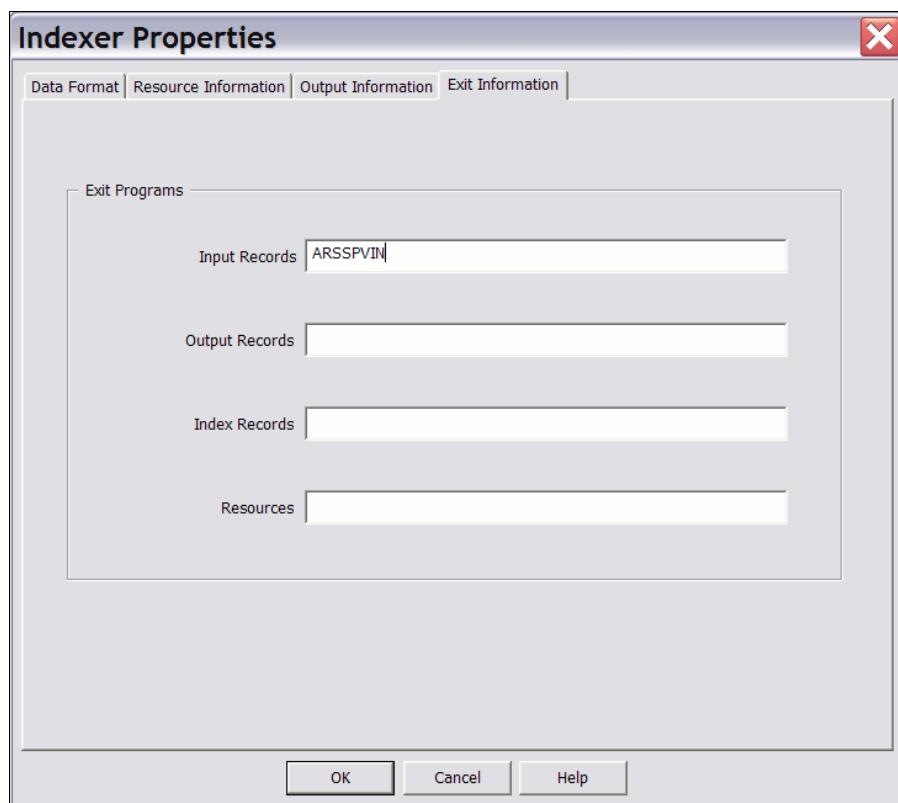


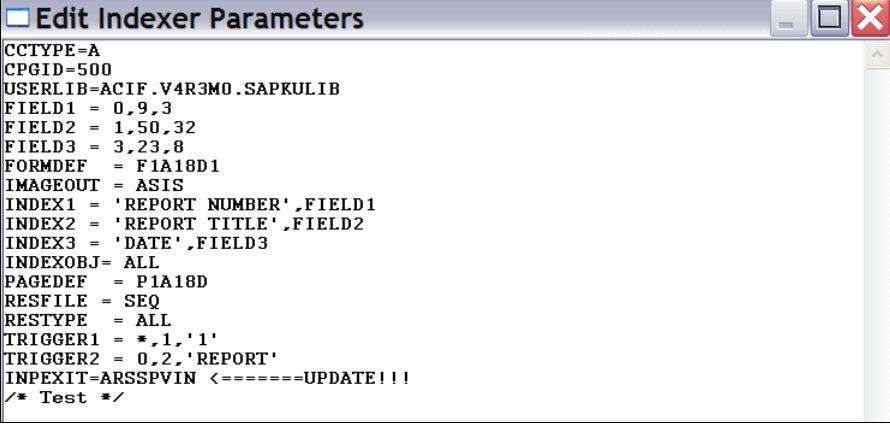
Figure 11-4 Specify Load Module Name in the Exit Information tab

4. In the Input Records field, enter the name of the exit.
5. Click **OK**.

The exit is added to your indexing parameters.

Editing the indexer parameters

Figure 11-5 shows the Edit Indexer Parameters window.



```
CCTYPE=A  
CPGID=500  
USERLIB=ACIF.V4R3M0.SAPKULIB  
FIELD1 = 0,9,3  
FIELD2 = 1,50,32  
FIELD3 = 3,23,8  
FORMDEF = F1A18D1  
IMAGEOUT = ASIS  
INDEX1 = 'REPORT NUMBER',FIELD1  
INDEX2 = 'REPORT TITLE',FIELD2  
INDEX3 = 'DATE',FIELD3  
INDEXOBJ= ALL  
PAGEDEF = P1A18D  
RESFILE = SEQ  
RESTYPE = ALL  
TRIGGER1 = *,1,'1'  
TRIGGER2 = 0,2,'REPORT'  
INPEXIT=ARSSPVIN <=====UPDATE!!!  
/* Test */
```

Figure 11-5 Edit exit information

If an application exists, edit your indexing parameters and add the following line, as in shown in Figure 11-5:

INPEXIT=ARSSPVIN

For more information about activating this exit, see the *Content Manager OnDemand for z/OS Version 9.0 Administration Guide*, SC19-3364.



Part 3

Advanced system concepts and design

This part contains the following chapters:

- ▶ Chapter 12, “Scalability, reliability, and availability architectures” on page 283
- ▶ Chapter 13, “Performance” on page 297



Scalability, reliability, and availability architectures

IBM Content Manager OnDemand (Content Manager OnDemand) is a lightweight process, that is, the Content Manager OnDemand code itself does not require extensive system resources to perform the functions that are required of it. Content Manager OnDemand installations scale to handle both large quantities of data and many users. The total quantity of data that is stored or retrieved at any time is the main contributor to the resource consumption on the server. This chapter focuses on the scalability, reliability, and availability of Content Manager OnDemand systems.

In this chapter, we cover the following topics:

- ▶ Scalability, reliability, and availability defined
- ▶ Scaling a Content Manager OnDemand system
- ▶ High availability

12.1 Scalability, reliability, and availability defined

This section defines scalability, reliability, and availability and how they relate to a Content Manager OnDemand system.

Scalability

Scalability is the ability of a Content Manager OnDemand system to handle a growing amount of work with no performance degradation. A Content Manager OnDemand system's performance improves with the addition of hardware and network resources and therefore is defined as a scalable system. Two types of scalability are defined:

- ▶ *Horizontal scalability* (or scale out): This type of scalability is achieved by adding more nodes, systems, or logical partitions (LPARs) to a Content Manager OnDemand instance. An example of horizontal scalability is adding more object servers to a Content Manager OnDemand instance.
- ▶ *Vertical scalability* (or scale up): This type of scalability is achieved by adding more resources to a single node in a Content Manager OnDemand instance. Typically, this type of scalability involves faster processors, more processors, memory, disks, or networking hardware.

Content Manager OnDemand is both horizontally and vertically scalable.

Reliability

Reliability is the ability of Content Manager OnDemand to perform and maintain functionality during regular workloads and during peak workloads. Peak workloads might occur regularly (for example, when everyone signs on at 9:00 a.m.) or periodically (at the end of the month when more processing than usual occurs). Or, peak workloads might occur sporadically (for example, when a special event occurs, such as a sales drive that results in more users using the system).

Availability

Availability is a measure of the time that a Content Manager OnDemand server or process functions normally, and a measure of the time that the recovery process requires after a component failure. It is the downtime (unavailability) that defines system availability. Availability is the amount of system uptime when the system is fully functional and accessible by all users.

Availability requires that the system provides a degree of redundancy to eliminate single points of failure (SPOFs). The greater the redundancy that is provided, the higher the availability of the system. A single physical machine is still a SPOF. For this reason, a high availability system topology typically involves horizontal scaling and redundancy across multiple machines.

High availability

High availability implies that no human intervention is needed to restore operation if a failure or outage occurs. A highly available system has an availability limit of at least 99%, which allows an average of 15 minutes each day to perform maintenance tasks (during which period the system is inaccessible to users). The degree of high availability that is achieved is a function of the amount of redundancy within the system and the degree to which this redundancy is automatically enabled.

Basically, two redundancy techniques are available:

- ▶ Passive redundancy: This redundancy is achieved by including enough excess capacity in the design to accommodate a performance decline, such as two Content Manager OnDemand servers (known as ARSSOCKD on z/OS and Multiplatforms) that access the same system tables and archive. If one server fails, the other server is available to take on the workload.
- ▶ Active redundancy: This redundancy is used to achieve high availability with no performance decline. In this case, at least double the required resources are allocated to the Content Manager OnDemand system. For example, if the peak workload requires 1.5 Content Manager OnDemand servers, three Content Manager OnDemand servers are configured to work in parallel. If one of the servers fails, the other two servers can take on the full workload with no performance degradation.

Systems typically become unavailable because of the lack of one or more of the following activities:

- ▶ Change control procedures (a failure to implement the appropriate procedures from installation verification through performance testing before you place the system into production)
- ▶ Monitoring of production system components (including total system workload, hardware, and network issues)
- ▶ Implementing high availability solutions (redundant systems and network connections)
- ▶ A comprehensive backup (and restore) process that is tested on a routine basis

A cost exists to implementing highly available high-performance systems. This cost must be weighed against the cost of not implementing such systems.

The following sections provide more information about example system implementations that allow high performance, scalability, reliability, and availability.

12.2 Scaling a Content Manager OnDemand system

A Content Manager OnDemand instance can be scaled from a single system image that performs all of the required tasks (data loading, library storage, and object storage) to a multiple system/multiple LPAR configuration, which offers higher levels of performance and availability. When a Content Manager OnDemand instance is distributed among multiple systems, these systems might be configured in the following ways:

- ▶ Single technology systems: The Content Manager OnDemand instance consists of systems that are of the same architecture. For example, all systems might be AIX systems.
- ▶ Multiple technology systems: The Content Manager OnDemand instance might consist of systems of different architectures. For example, the library server and an object server might be on a z/OS system; two other object servers might be on AIX systems; and another object server might be on a Microsoft Windows system.

In both of these scenarios, the configuration results in a single Content Manager OnDemand instance view from both the administrative and user perspectives.

With this flexibility and scalability, you can configure Content Manager OnDemand systems so that they meet a wide range of both workload and operational requirements.

Examples of these configurations are illustrated in the figures in this section. These figures are only a sample of the possible configurations that are used to illustrate the basic scalability features.

Figure 12-1 illustrates a single Content Manager OnDemand instance. In this figure, the Content Manager OnDemand server supports the library server, one or more object servers, and one or more load processes. The following sections provide examples of how the Content Manager OnDemand server can be scaled both vertically and horizontally.

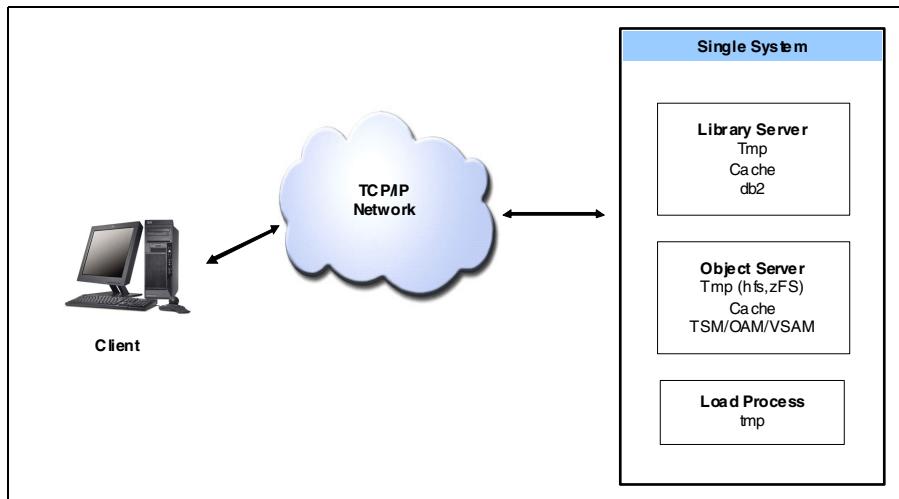


Figure 12-1 Scalability: A single instance (simple client/server) setup

12.2.1 Vertical scalability

You can scale Content Manager OnDemand vertically by expanding the system, by using a larger system, through application design, or through parallel archive access.

Expanding the system

Content Manager OnDemand is vertically scalable if the system that it is running on is scalable. Vertical scalability is achieved by adding more hardware to the system. This hardware might be in the form of faster processors, more processors, memory, disks, I/O, or network capacity.

The limit to the amount of possible vertical scalability is the architectural hardware constraints of the system. For example, if the system supports only 24 GB of memory, that memory limitation can be overcome only by buying a larger system.

Using a larger system

You can scale a Content Manager OnDemand system vertically by using a larger system in one of two ways:

- ▶ Installing a larger system within the same family and architecture, for example, moving from an entry-level AIX system to an enterprise-level AIX system
- ▶ Installing a larger system from a different architecture and family, for example, moving a Content Manager OnDemand server from a Windows system to an AIX system

Application design

Modern computer systems contain multiple cores. They can perform multithreaded processing. Modern computer system operating systems allow parallelism in operations. To take advantage of these hardware and software features, an application must be designed so that it can run in parallel at multiple levels. Content Manager OnDemand can take advantage of both.

At the process level, the Content Manager OnDemand server runs multiple processes:

- ▶ A library server
- ▶ One or more object servers
- ▶ One or more load jobs
- ▶ The expiration process

At the thread level:

- ▶ The library server is designed so that it is multithreaded and it can service multiple incoming data requests on different threads and perform multiple database queries in parallel.
- ▶ The object server is also multithreaded so that multiple users can concurrently retrieve data from the Content Manager OnDemand archive.

Parallel archive access

When you access the Tivoli Storage Manager or object access method (OAM) archives, a store or retrieve request is sent to Archive Storage Manager (ASM). ASM then either stores or retrieves the data and returns the result to the Content Manager OnDemand server. If this process is conducted in a serial fashion, the archive storage access mechanism becomes a bottleneck at high transaction rates. To overcome this potential bottleneck, Content Manager OnDemand implements connection pooling to the storage archives.

Content Manager OnDemand maintains a pool of connections to the archive. When an archive store or retrieve request is received, an available connection from the pool is selected to perform the request. This connection allows both faster access to the archive (by eliminating the start process each time a connection is requested) and for the parallel execution of the store or retrieve operations.

On IBM i, when you access the ASM archives, connection pooling is not required for store requests. When a store request is made, ASM opens a connection and keeps it open until the data store request is complete. In addition, ASM allows the aggregation of objects, sending fewer objects to storage media than otherwise are sent without aggregation.

On Multiplatforms and z/OS, you can aggregate documents that are loaded from Content Manager OnDemand Web Enablement Kit (ODWEK) before you store them in the archive. The document is stored to cache where it is appended to the storage object until the object reaches 10 MB (defined storage object size), at which point it is migrated to a storage manager, such as Tivoli Storage Manager. For more information about this topic, see the following website:

<http://www.ibm.com/support/docview.wss?uid=swg21587507>

12.2.2 Horizontal scalability: Library server

Even though Content Manager OnDemand allows a single library server for each instance, this library server can be scaled horizontally. The library server is scaled horizontally by using one or both of the following methods:

- ▶ The database tables (both the system and the application group) can be placed in different databases (z/OS) or different table spaces (Multiplatforms and z/OS) at the table level. Therefore, each of these tables can scale to the maximum practical size that is supported by the database within the operational constraints of maintenance and performance. Content Manager OnDemand does not impose a limitation.
- ▶ The application group data table design facilitates the following actions:
 - You can create as many application groups as you need to support the required data to be archived.
 - Each application group can be segmented into multiple tables where the table segmentation is based on size.
 - Each of these application group data tables can be placed in a separate database (z/OS) or table space (Multiplatforms and z/OS).

12.2.3 Horizontal scalability: Multiple object servers

For Multiplatforms and z/OS, you can scale a Content Manager OnDemand system horizontally by using multiple object servers.

In the example that is shown in Figure 12-2, the Content Manager OnDemand system is horizontally scaled by placing the library server, object servers, and load processes on multiple systems.

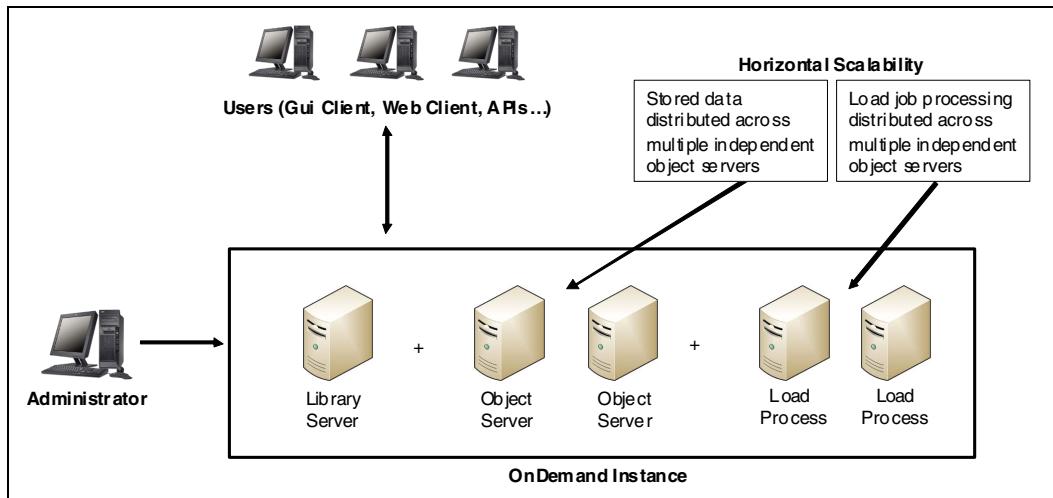


Figure 12-2 Horizontal scaling: Multiple object servers (z/OS and Multiplatforms)

This form of horizontal scalability provides better performance, reliability, and scalability by distributing the storage and retrieval workload over multiple systems.

From a Content Manager OnDemand perspective, no limit exists to the number of object and load process servers. Each of the servers can run to its maximum capacity. Operational limitations are imposed by the TCP network bandwidth that connects all of the servers and by the available data center floor space. Both of these constraints can be reduced by placing multiple servers in a rack-mounted configuration.

This example and all of the following examples, from an external perspective, show a single Content Manager OnDemand instance. The fact that the system consists of multiple distributed systems is transparent to both of the following groups:

- ▶ The Content Manager OnDemand administrator, who continues to administer the system through the Content Manager OnDemand Administrator Client as though it is a single physical system.
- ▶ The Content Manager OnDemand users, who continue to access the whole system through a single IP address (that of the library server) and see only a single system from their perspective.

12.2.4 Horizontal and vertical scalability: Storage manager

This form of horizontal scalability provides better performance, reliability, and scalability by distributing the storage and retrieval workload over multiple storage subsystems within each object server.

An object server controls the storage and retrieval of the archived data. The archived data is stored in a storage subsystem. The number and architecture of these subsystems can be scaled to the limitations of the subsystem. Each object server can support one or more storage subsystems, and each storage subsystem can consist of multiple storage devices, as shown in Figure 12-3.

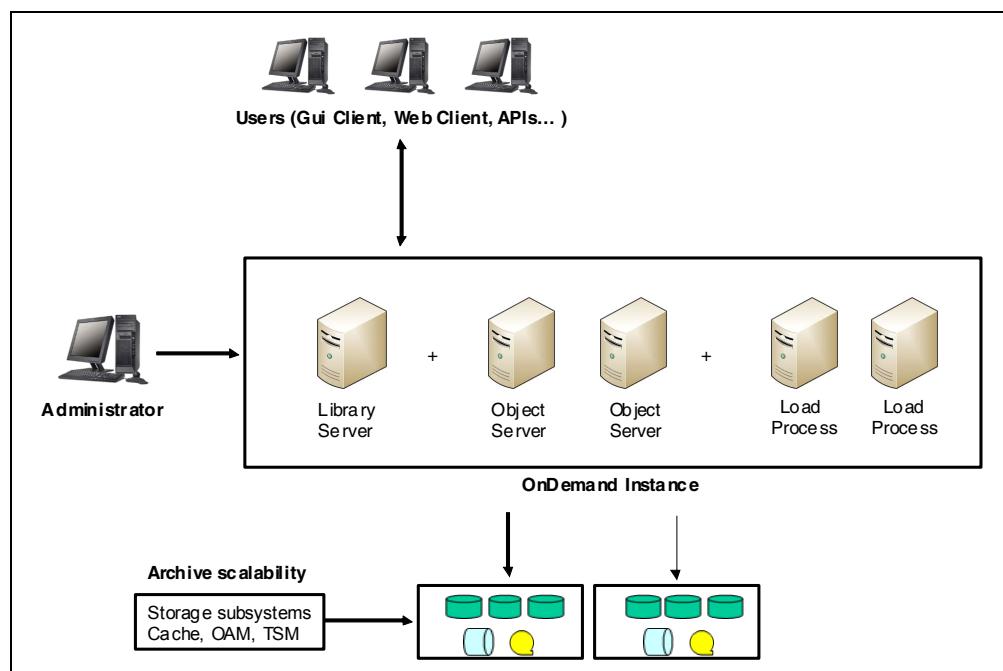


Figure 12-3 Horizontal and vertical scaling: Multiple storage managers

Each object server can have multiple storage subsystems of different types:

- ▶ Cache: The cache storage subsystem is controlled directly by the object server. Data is written to and read directly from cache. Cache consists of one or more cache file systems. Each cache file system can be mounted on a different device in its own directory. Each device can be placed on its own independent I/O interface/channel. Content Manager OnDemand does not impose a limit on the number of devices.
- ▶ IBM Tivoli Storage Manager: Tivoli Storage Manager is an archive storage subsystem. The Content Manager OnDemand object server sends data to and requests data from Tivoli Storage Manager. Each Tivoli Storage Manager server can be installed on its own system (for example, an AIX server). The Content Manager OnDemand object server allows the connection of multiple Tivoli Storage Manager servers. So, for example, if the Content Manager OnDemand object server is an AIX system and the data that is managed by that object server is stored in three Tivoli Storage Manager archives (all of which are AIX systems), the total processing capacity for that object server is four AIX systems. Each of the AIX systems can be configured with as many processors, memory, disks, and I/O as needed, up to its architectural limitation. If more capacity is needed, more Tivoli Storage Manager servers or object servers can be added.
- ▶ Object access method (OAM): OAM is a z/OS archive storage subsystem only. Only one OAM archive exists for each system. Scalability within the archive is achieved by increasing the number of storage groups. A z/OS system can grow by increasing the number of processors, amount of memory, number of disks, and I/O. If more capacity is needed than can be provided by a single system, z/OS allows multiple systems to be connected in a Parallel Sysplex. All of these systems then can access the same OAM subsystem, therefore providing unparalleled scalability, reliability, availability, and performance.

Both Tivoli Storage Manager and OAM provide hierarchical data management facilities. Data can be stored on different devices based on the age or predicted frequency of data access. For example, frequently accessed data might be placed on high-speed disk and infrequently accessed data might be placed on tape. When the data is requested by a user, the location of the data is transparent to the user. The only perceived difference from a user perspective is the response time, which is mainly a factor of the type of device on which the data is stored. In this example, tape access is slower than disk access.

In summary, better performance is achieved by distributing the storage and retrieval workload over multiple systems and multiple devices.

12.2.5 Horizontal scalability: Multiple logical partitions and systems

This scenario is similar to the multiple object server scenario where each object server is running on a separate system. In this case, the library server and one or more object servers are installed in separate LPARs on one or more physical systems, as shown in Figure 12-4 on page 291.

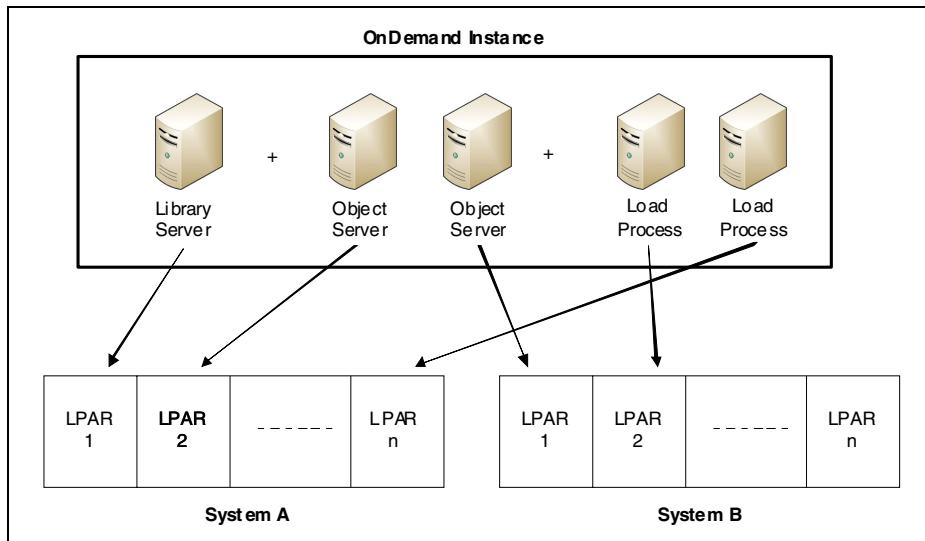


Figure 12-4 Horizontal and vertical scaling with multiple LPARs

This scenario is in organizations with large systems, such as AIX or z/OS, that are installed and that have enough available capacity to support the required Content Manager OnDemand workload. One advantage of this configuration is that you can control the priority of work and computer resource distribution to each of the LPARs, such as the number of processors or the processing priority (depending on the computer system/operating system architecture) that is allocated to each of the LPARs. So, for example, load jobs can be assigned a low priority during the day when the focus is on data retrieval and a high priority during the night when the focus is on data loading.

This setup supports horizontal scalability by using multiple technologies as appropriate. The main constraint is that clients must have access to all systems through TCP/IP.

12.2.6 Multiple server configuration rules

The following general rules apply when you configure multiple Content Manager OnDemand servers. In all cases, for additional guidance, see the appropriate Content Manager OnDemand documentation or contact Content Manager OnDemand Lab Services.

- ▶ Each Content Manager OnDemand server has its own set of configuration files.
- ▶ The parameters in all configuration files must be set so that all of the servers are part of the same instance.
- ▶ The Content Manager OnDemand clients connect to the IP address listening port of the Content Manager OnDemand server (library server module).
- ▶ The documents are retrieved from the various object servers based on the location information that is returned by the library server. This retrieval is transparent to the client systems.
- ▶ Parallel load processes must have separate temp directories.

Figure 12-5 on page 292 depicts this configuration type.

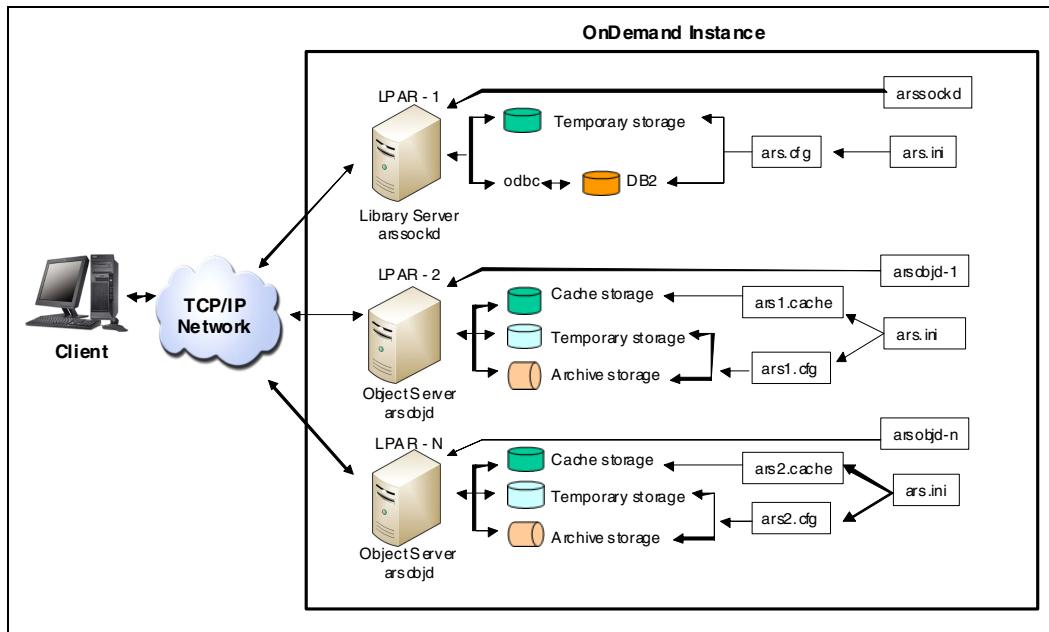


Figure 12-5 Multiple server configuration

12.3 High availability

The concept of high availability roughly equates to a system and its data being available (accessible by users) almost all of the time, 24 hours a day, 7 days a week, and 365 days a year. In actuality, 100% availability is not a cost-effective reality today for most implementations; rather, it is a goal. The goal is to design and build systems that are highly available by minimizing both planned and unplanned outages that can be caused by SPOFs.

12.3.1 Redundant systems: All platforms

Various techniques are employed on all platforms to achieve near high availability. These techniques are based on creating as much redundancy as possible within the system and the data that the systems include:

- ▶ Preventing data loss: Employing various levels of RAID to store the data on disk.
- ▶ Duplicating the data: Creating near real-time copies of the data on backup devices that replace the online devices if the online devices fail.
- ▶ Duplicate systems: A duplicate system (hardware, software, and data) is maintained (either locally or remotely), and when the main system fails, users are automatically directed to the duplicate system.
- ▶ Network redundancy: Creating multiple paths through the network so that if one path (or router) fails, the network continues to function.

All of these techniques work well and provide various levels of near real-time high availability based on the degree to which the redundant systems are created and are kept in active-standby mode.

12.3.2 Multiple LPAR sysplex: z/OS

The z/OS operating system has a high availability architecture that is built into it. A z/OS Parallel Sysplex is a tightly coupled cluster of independent z/OS systems that connect through an Internet Protocol network. A cluster is 2 - 32 independent systems that are locally or geographically dispersed. Communication between the z/OS systems in the sysplex is handled through the cross-system Coupling Facility (XCF). A z/OS Parallel Sysplex implementation provides the highest level of high availability in the industry.

Figure 12-6 illustrates a Content Manager OnDemand implementation of a two-system highly available z/OS sysplex system.

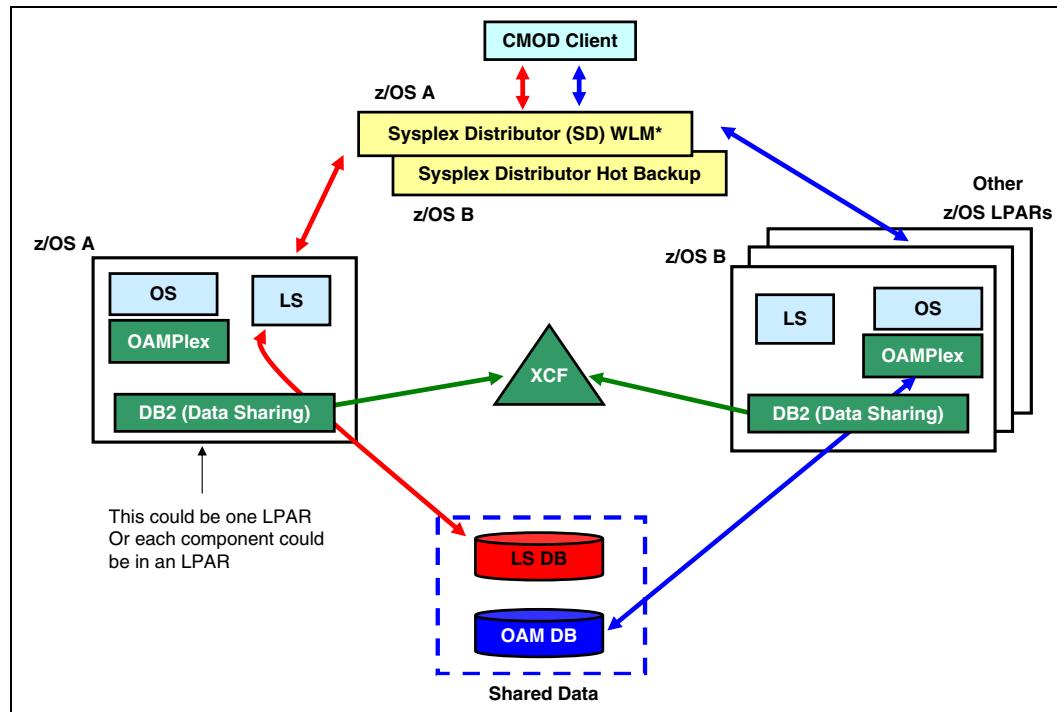


Figure 12-6 Scalability: Parallel Sysplex/multiple LPARs (z/OS)

Figure 12-6 illustrates an example of a two-system Content Manager OnDemand Parallel Sysplex implementation. z/OS system A contains a library server and an object server. The library server and object server can be either combined in a single executable file (most common z/OS implementation) or separated into two executable files, in which case they are installed in separate LPARs. z/OS system B shows a multiple LPAR system with a combined library/object server that is installed in each of the LPARs.

Both of these systems (all LPARs and all instances of the Content Manager OnDemand server) access a single set of Content Manager OnDemand database tables through DB2 data sharing. They also access a single OAM archive system through an OAMplex. Not shown in the figure is the access to a single Job Entry Subsystem (JES) spool and a shared file system (which consists of a set of hierarchical file systems (HFS) or z/OS file systems (zFS)). The term “single” is used to imply that the same set of data is available to all systems concurrently. Each of these single systems consists of highly redundant components and therefore do not represent a SPOF.

The z/OS Parallel Sysplex technology enables the Content Manager OnDemand servers to share configuration files, database, JES, HFS, and archive. For performance reasons, all HFS read/write directories that are used for temporary storage of data are configured as unique to each Content Manager OnDemand server.

From a client perspective, the “cluster” is a single IP address. Incoming client requests are received by the sysplex distributor/Workload Manager (WLM). WLM monitors the various systems in the Parallel Sysplex and selects the appropriate Content Manager OnDemand server to forward the request to based on the current system workload and availability, so that the system that is more available (less busy) receives the request.

12.3.3 High availability: IBM i

IBM PowerHA® SystemMirror® for i is the integrated IBM storage-based clustering solution for high availability and disaster recovery. The data and applications are deployed into storage pools, which are called *independent auxiliary storage pools* (IASPs). IASPs can be deployed by using either internal or external storage. At any time, the nodes in the cluster can switch roles and become either a primary or backup node. PowerHA SystemMirror can be used for on-demand role swap operations.

The IBM Power Systems™ Capacity BackUp (CBU) offerings support disaster recovery and high availability needs. The CBU offerings recognize that true high availability or disaster recovery solutions require at least two systems. If one system is not available, the other system takes over. The CBU offering provides flexible and economic options for deploying business continuity operations.

In a high availability environment on IBM i, you might not want to replicate the following directories because OnDemand places only temporary data in them, and this data might occupy a large amount of space:

- ▶ Do not replicate the temporary integrated file system (IFS) directories for your instances. For example, do not replicate /QIBM/UserData/OnDemand/*QUSROND*/TMP or /QIBM/UserData/OnDemand/*QUSROND*/PRTTMP, where *QUSROND* is your instance name.
- ▶ Do not replicate the home directory for the user that is storing data. For example, if JOHNDOE is the name of the user profile that stores data into Content Manager OnDemand, do no replicate /home/JOHNDOE.
- ▶ Do not replicate the /tmp directory.

12.3.4 Horizontal and vertical scalability summary

You can use the architectural flexibility of Content Manager OnDemand (Figure 12-7 on page 295) to select the correctly sized system based on your needs. A Content Manager OnDemand implementation can be scaled both vertically (by using larger and larger systems) and horizontally (by increasing the number of systems that are part of the Content Manager OnDemand instance).

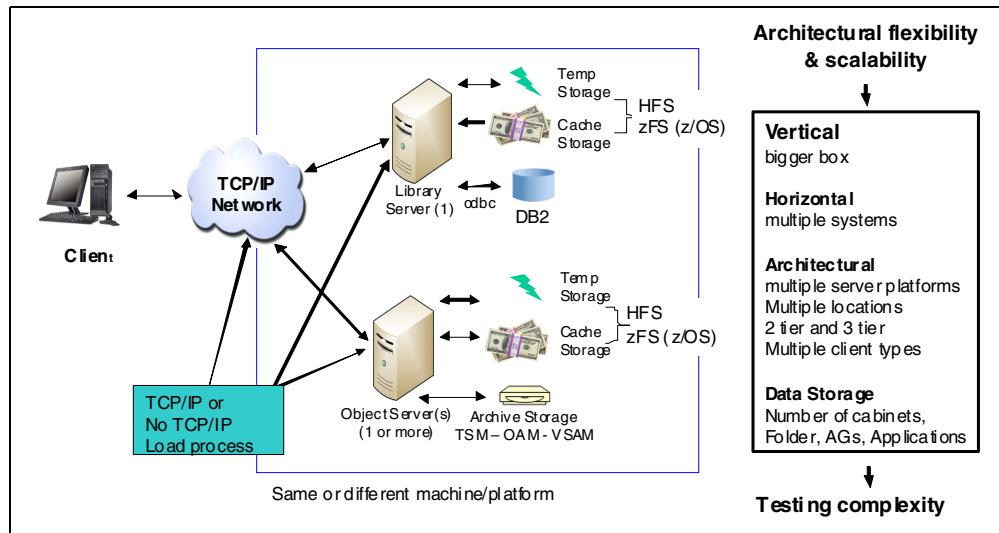


Figure 12-7 Horizontal and vertical scalability

A Content Manager OnDemand server can scale from a Windows server up to a cluster of z/OS systems. It is important to initially select an installation that meets the following requirements:

- ▶ Appropriate for your current workload in terms of the following items:
 - Performance
 - Reliability
 - Availability
 - Scalability
- ▶ Support for your future growth requirements if the following actions are necessary:
 - Increase the number of users that access the system
 - Increase the quantity of data that is stored in the system
- ▶ Change in the types of archived data
- ▶ Change in the preprocessing requirements



Performance

In this chapter, we describe the ways in which the various components within IBM Content Manager OnDemand (Content Manager OnDemand) might be configured or tuned to enhance performance. In most cases, it is not possible to give specific parameter values; however, we provide broad concepts and recommendations in areas where tuning for performance is possible.

In this chapter, we cover the following topics:

- ▶ Tuning Content Manager OnDemand to enhance performance
- ▶ Data loading performance
- ▶ Data retrieval performance
- ▶ Performance issues that are based on data type

13.1 Tuning Content Manager OnDemand to enhance performance

Two components make up performance: throughput and response time:

- ▶ Throughput: The number of transactions (Content Manager OnDemand requests) that can be satisfied for each unit of time. The more transactions that are run for each unit of time, the higher the throughput. Higher throughput implies that more users can be served concurrently and more load jobs can be run in parallel. If the throughput values are low, the system might not be able to support the required number of users.
- ▶ Response time: The amount of time it takes to service a single transaction (Content Manager OnDemand request). Faster response times imply that the users are able to retrieve their data faster from the archive, which in turn leads to more satisfied users. If the response time is slow, users are dissatisfied with the system.

A high performance system, such as Content Manager OnDemand, provides both high throughput and short response times.

The following sections describe the various components of a Content Manager OnDemand system and its architecture. They provide guidance about the parameters and configurations that you can change to improve performance.

The ability to separate the object server from the library server offers two main advantages:

- ▶ The ability to share workload by dedicating machines to individual tasks
- ▶ The ability to reduce the impact of retrieving a large piece of data over a network that is either slow or overloaded

13.1.1 Content Manager OnDemand configuration

How reports are defined, indexed, and stored within Content Manager OnDemand greatly influences the speed at which Content Manager OnDemand can retrieve them. Various hints and tips for the optimum way to define reports within Content Manager OnDemand are described in Chapter 3, “Administration” on page 45.

13.1.2 System logging

Use Content Manager OnDemand system logging for usage monitoring, charge-back, or troubleshooting. Because system logging involves writing all of the selected log messages to disk, you incur an increase in both resource usage and response time. Logging increases both the amount of the processor that is used and the amount of I/O to disk. For this reason, select only the types of logging that you want performed for a particular application group. Depending on your system usage requirements, you might decide to perform any of the following tasks:

- ▶ Turn off all system logging.
- ▶ Record a minimal amount of information (only the information that is needed for reporting functions).
- ▶ Record all transactions.
- ▶ Record the log information to one or more external files by using the system log exit.
- ▶ Turn on system logging only while you troubleshoot the system.
- ▶ Turn on system logging once every time period to sample the system usage patterns.

13.1.3 System management

For effective system management, set the appropriate value for **ARS_NUM_DBDRV** and create the correct file systems for various Content Manager OnDemand components.

ARS_NUM_DBDRV

The **ARS_NUM_DBDRV** parameter is set in the `ars.cfg` file. This parameter is the maximum number of threads that are concurrently open between the Content Manager OnDemand library server and DB2. Typically, this value is set to a number between 4 - 30. This number must be large enough to support all of the concurrent database requests from all users and clients and Content Manager OnDemand commands and daemons, such as **ARSLOAD**, **ARSDOC**, **ARSDB**, **ARSMINT**, and **ARSADMIN**. This number must not exceed the number of DB2 batch connections (**MAXDBATS** for z/OS and **MAXAPPLS** for Multiprocessing (MP)). The number of DB2 batch connections must be greater than the **ARS_NUM_DBDRV**, plus all of the other connections that are required by all DB2 applications that you defined in your DB2 configuration.

For systems that are running several large load jobs in parallel, or for systems with large numbers of active users, increase this parameter from the default of 4.

File systems on UNIX

During the installation and setup of Content Manager OnDemand, one of the tasks is to create the file systems that are required to contain the various Content Manager OnDemand components.

For performance reasons, when the Content Manager OnDemand file systems are created, the following components *must not* be on the same physical media:

- ▶ Cache file system
- ▶ Database file system
- ▶ Primary logs file system
- ▶ Secondary logs file system
- ▶ Load/indexing file system
- ▶ Content Manager OnDemand temporary space file system

13.1.4 Storage management

Regardless of the platforms, storage management with Content Manager OnDemand can be divided into two areas: cache storage that is managed by Content Manager OnDemand and archive media that is managed by an external product, such as IBM Tivoli Storage Manager, object access method (OAM), Virtual Storage Access Method (VSAM), or Archive Storage Manager (ASM).

For effective storage management, one key performance feature of Content Manager OnDemand is its ability to load data to archive media, while simultaneously retaining a temporary cached copy of the most recent archived data on fast access media (such as the hard disk drive (HDD)). Content Manager OnDemand handles the expiration and management of this cached copy of the data. After a certain predefined period elapses, the data is removed from cache. The only remaining copy is held on the much slower archive media that is managed by either Tivoli Storage Manager, OAM, VSAM, or ASM, depending on the platform.

If performance problems are encountered at the storage manager level, the issue is almost always related to the inherent qualities of the slower media types (such as optical platters and tape volumes) or how the archive media manager is configured.

Several of the parameters that affect storage management are **ARS_NUM_OAMSRVR**, **ARS_NUM_OAMSRVR_SLOW_RETRIEVE**, and **ARS_OAM_SLOW_RETRIEVE_THRESHOLD**.

ARS_NUM_OAMSRVR

This parameter specifies the maximum number of concurrently attached threads between the Content Manager OnDemand object server and OAM for z/OS. Typically, this value is set to a number between 4 - 30, depending on client access patterns and object storage locations (disk versus tape). This parameter has a maximum value of 30. Any value larger than 30 result in a U0039 abend.

ARS_NUM_OAMSRVR_SLOW_RETRIEVE

This parameter determines the number of task control blocks (TCBs) that the Content Manager OnDemand server starts to handle connections to OAM for retrievals from objects with a slow retrieval time as defined by the **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** parameter. The **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** parameter applies to all object servers. If the value that is specified for this parameter is zero (0), no TCBs are dedicated for slow retrievals. All retrievals are processed by the TCBs that are associated with the **ARS_NUM_OAMSRVR** parameter. The default is zero (0). The **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCBs are in addition to the **ARS_NUM_OAMSRVR** TCBs, and they use additional DB2 connections.

ARS_OAM_SLOW_RETRIEVE_THRESHOLD

This parameter specifies the threshold at which OAM retrievals are processed by the TCBs that are associated with the **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** parameter. If the estimated retrieval time for an object (as indicated by QELQERRT) is greater than or equal to the value of the **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** parameter, the OSREQ RETRIEVE is processed by an **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCB. The default value is 12000. For other valid QELQERRT values, see the *Object Access Method Application Programmer's Reference*, SC35-0425-08. An **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** value of zero (0) with a nonzero **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** value causes all OAM retrieve requests to be processed by the **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCBs, while the **ARS_NUM_OAMSRVR** TCBs process store, query, and delete requests.

13.2 Data loading performance

The data loading process is illustrated in Figure 13-1 on page 301. The process begins with the Content Manager OnDemand Administrator Client defining the application group and application parameters for the reports to be loaded. These parameters are stored in the Content Manager OnDemand system tables on the library server.

Data Loading

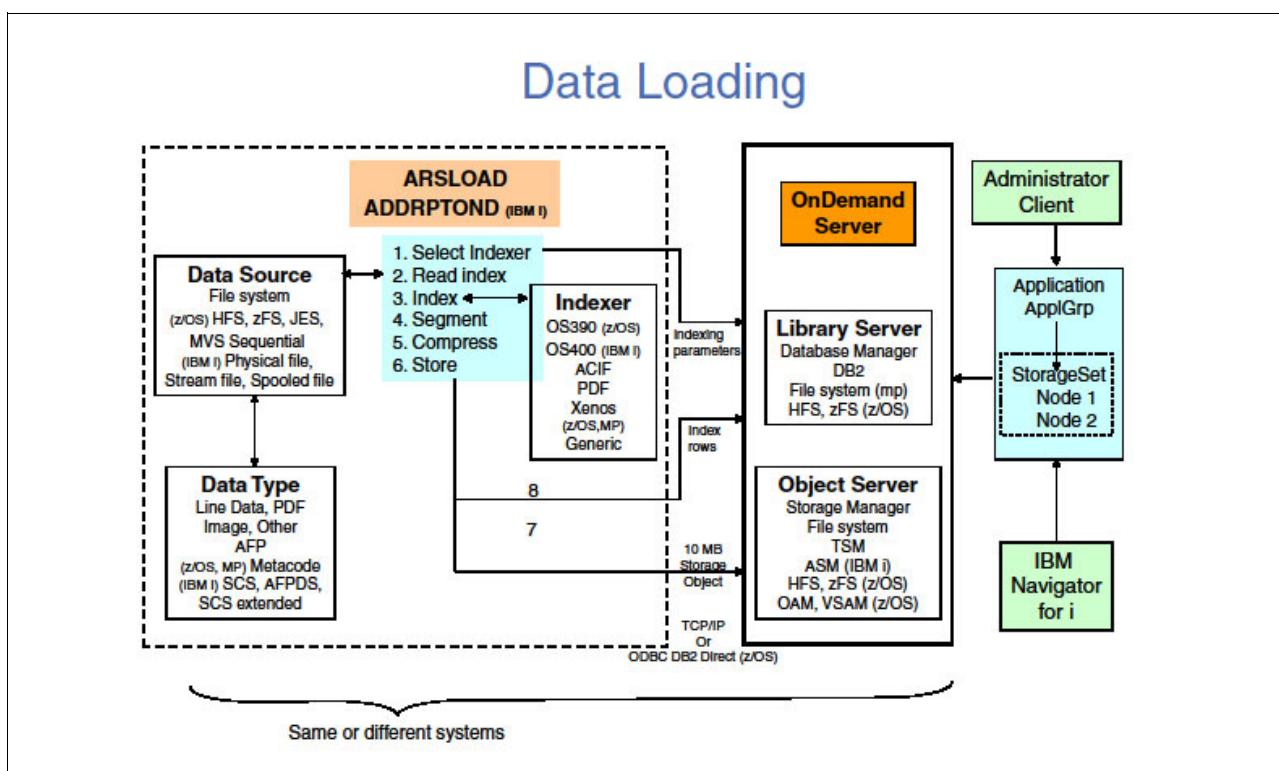


Figure 13-1 Data loading

During the load process, in addition to any command-line parameters that are supplied, the application group and application parameters are retrieved from the library server. Then, based on the parameter definitions, the load process completes the following steps:

1. Selects the indexer to be used for indexing the report data and retrieves the indexing parameters.
2. Reads in the report data from the identified source location. The input report data can be of any data type.
3. Indexes the report data based on the defined indexing parameters.
4. Segments the report into “documents”.
5. Compresses the documents.
6. Stores the compressed documents in storage objects (10 MB by default).
7. Sends the storage objects to the object server where they are stored in the identified archive (storage node).
8. Sends the index data for the stored objects to the library server where the indexes are stored in the appropriate application group data table.

13.2.1 Factors that affect the load performance

Many factors affect load performance:

- Quantity, speed, and capacity of the available hardware (processors, memory, disks, I/O channels, network, and so on)
- Network bandwidth and throughput
- Operating system tuning components: DB2, TCP/IP, and Language Environment

- ▶ Content Manager OnDemand tunable components
- ▶ Storage management tunable components: UNIX System Services, z/OS file system (zFS), hierarchical file system (HFS), OAM, Tivoli Storage Manager, and ASM
- ▶ Data components:
 - Report file size, document file size (or in the case of large objects, report segment size), and number of documents per report.
 - Number and distribution of triggers, fields, and indexes per document.
 - Data type and required data conversion (if any).
 - Resource collection for AFP and Portable Document Format (PDF).
 - Document compressibility, which is a function of document data complexity and data type. Text (such as Line Data or SCS) is typically more compressible than AFP, which is typically more compressible than PDF.
 - Storage object size (10 MB default): Contains 100 KB compressed object, which contains a compressed document.
 - Exit routines/programs.

13.2.2 Recommendations

For the most optimal performance in loading, we recommend the following practices:

- ▶ For Multiplatforms and z/OS, run parallel load jobs to take advantage of multiprocessors, large memory pools, multiple data paths, and multiple disk drives.
- ▶ Ensure that each parallel load is loading to a different application group.
- ▶ Ensure that you set up a different temp directory for each of the parallel loads. The **-c indexDir** indexer parameter (which specifies the directory in which the indexer stores temporary data) must always be specified for **ARSLOAD** and must be unique for each running **ARSLOAD** process.
- ▶ For IBM i, start multiple output queue monitors over a single output queue to improve throughput and take advantage of multiprocessors, large memory pools, and multiple disk drives.
- ▶ Each Content Manager OnDemand process is limited by the performance of a single processor. For example, the OS/400 indexer uses only one processor when it indexes a document. Using two or more processors in your system or LPAR does not improve the performance of the OS/400 indexer. However, by using two or more processors in your system or LPAR, you might be able to run multiple load jobs simultaneously. You can start multiple output queue monitors over a single output queue to improve document load performance.
- ▶ For IBM i, the use of the Merge Spooled Files (**MRGSPLFOND**) command can provide significant performance improvements when you load SCS spooled files.

- ▶ For IBM i, depending on your retrieval patterns and system hardware configuration, it might be advantageous to *not* store a duplicate set of documents in the Content Manager OnDemand cache when you use ASM because ASM might already be using disk space. If the application group uses ASM, caches the data, and specifies the migration of data at load time, two copies of the data are stored during the load. One copy is stored in cache, and one copy is stored in the ASMREQUEST directory.

To avoid storing a duplicate set of documents in cache for non-APP data, change Cache Data to No on the Storage Management tab of your application group definition. To avoid storing a duplicate set of documents in cache for APP data, you might change Document Data to No Cache but leave Resource Data in cache for faster retrieval.

- ▶ For IBM i, every user that loads data must have a home directory. If users do not have a home directory, the temporary files are stored in the root directory of the integrated file system (IFS).
- ▶ If the data source is on a remote system, you can load the data into Content Manager OnDemand on the remote system and directly store the export data to the specified Content Manager OnDemand library and object server.

Or, if the data source is on a remote system, you also can upload the data to the specified Content Manager OnDemand server through FTP and then load the data on the selected Content Manager OnDemand system.

- ▶ For Multiplatforms and z/OS, all file systems must be dedicated file systems that are mounted on their own mount points.
- ▶ For z/OS, when you load PDF reports (by using the PDF Indexer), placing the input report in the HFS or zFS causes the load to run nearly 50 times faster than compared to the input report that is placed in a VSAM file.

13.2.3 Load testing

The goal of load testing is to verify that, under stressful system conditions, the required amount of data can be loaded into the Content Manager OnDemand system within a time window.

A general approach to load testing a system is described:

- ▶ Parallel loads: Run a single load and measure the load throughput. If the throughput does not meet the requirements, run two loads in parallel and measure the throughput. While the loads are run, collect system statistics to determine the system resources that are being used and any potential bottlenecks. Tune or acquire additional system resources as needed. Progressively increase the number of parallel loads until the required throughput is met.

Note: For most users, a single load process meets the ingestion throughput requirements.

- ▶ Data types and exits: A different data type, and whether an exit is started during the load process, affects the load throughput. Test samples of the different types that represent the general loads.

13.3 Data retrieval performance

All Content Manager OnDemand clients (such as the Windows client, CICS client, IBM Content Navigator, ODWEK application programming interfaces (APIs), and structured APIs) retrieve data from the Content Manager OnDemand server by using a standard proprietary Content Manager OnDemand protocol. From a Content Manager OnDemand server perspective, no difference exists between one client and another client.

13.3.1 Data retrieval parameters

Various parameters affect data retrieval performance.

Folder parameters: General tab

In the Content Manager OnDemand Administrator Client, under the Folder parameter and on the General tab, the following option is available:

- ▶ Note Search: If the Annotation flags in a document database are set to No in the Advanced tab of the General window of the Application Group, this option determines when Content Manager OnDemand searches the database for annotations and notifies users that annotations exist for the documents that match a query. Content Manager OnDemand provides three search and notification methods:
 - Hit List: Content Manager OnDemand searches for annotations when the user runs a query. When annotations exist for a document, the client programs display a note icon next to it in the document list. This method has a direct performance impact on the generation of the document list.
 - Retrieve: Content Manager OnDemand searches for annotations when the user selects a document for viewing. This method is the default and recommended value.
 - Note: Content Manager OnDemand searches for annotations when the user chooses the Note option while the user views a document.

Folder parameters: Permissions tab

In the Content Manager OnDemand Administrator Client, under Folder parameters and on the Permission tab, the following option is available:

- ▶ Max Hits: Determines the maximum number of hits that are retrieved and transmitted to the client. By reducing the maximum number of hits, users are forced to enter queries that better match the documents that they are searching for. By reducing the maximum number of hits, the system resources are used optimally both in performing the queries and in downloading the resulting document list.

TCP/IP considerations

A known Windows configuration setting might affect performance when you connect to a Content Manager OnDemand server. During repeated searches and retrievals on a Content Manager OnDemand server, many Windows sockets are opened and closed. Two default Windows settings might affect heavy traffic between the client and the Content Manager OnDemand server:

- ▶ When an application closes a Windows socket, Windows places the socket's port into TIME_WAIT status for 240 seconds; during this time, the port cannot be reused.
- ▶ Windows limits the number of ports that an application can use to 5000.

To avoid the problems that might result, change the values for the timeout wait time and number of ports by editing the Windows registry:

- ▶ Change the value of the timeout wait time from 240 seconds to a lower number (valid values are 30 - 300 seconds). The key's name is shown:
HKEY_Local_Machine\System\CurrentControlSet\services\Tcpip\Parameters\TcpTimedWaitDelay
- ▶ Increase the maximum port number from its default of 5000 to a higher number (valid values are 5000 - 65534). The key's name is shown:
HKEY_Local_Machine\System\CurrentControlSet\services\Tcpip\Parameters\MaxUserPort

For more information about TcpTimedWaitDelay and MaxUserPort, see your Windows documentation.

Verify with your network personnel that you are setting the values that are appropriate for your environment correctly.

13.3.2 Factors that affect retrieval performance

Figure 13-2 shows the data retrieval performance testing, which is an illustration of the methodology that is used by the Content Manager OnDemand lab for its internal performance testing. On the client side (where the cTest program is), both throughput and response time are recorded. The definitions for throughput and response time are shown:

- ▶ Throughput: The amount of work that is performed over a period of time (How many transactions can the Content Manager OnDemand server (CMOD SERVER in the figure) run at the same time?)
- ▶ Response time: The time that is elapsed between when a request is submitted and when the response from that request is returned (How long does it take for a transaction to run?)

Maximizing performance is a balancing act between optimizing throughput (which is based on keeping the computing resources busy) and optimizing response times (which requires the computing resources to be available when they are needed). As the throughput increases, so does the response time.

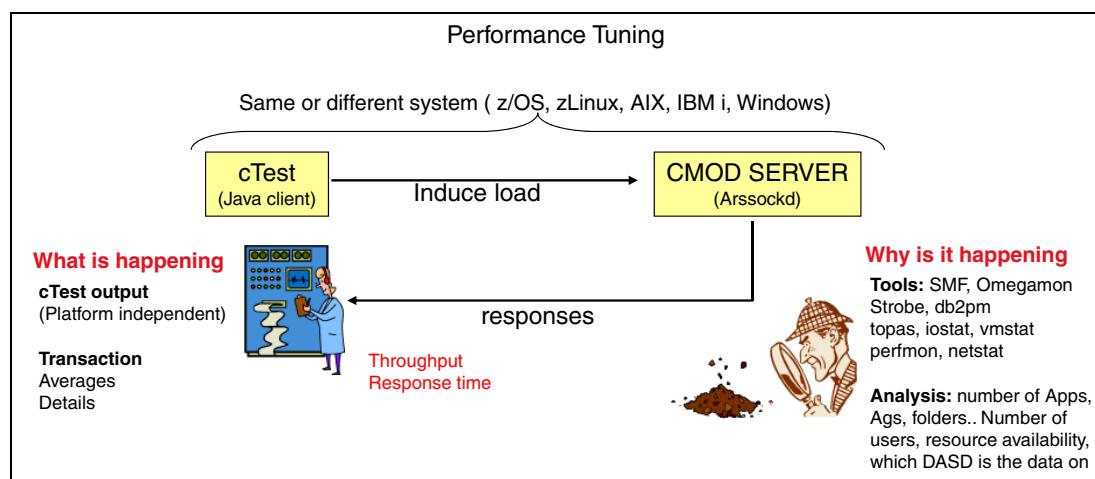


Figure 13-2 Data retrieval performance testing

The concepts that are shown in Figure 13-2 are described for your reference.

The retrieval performance is mostly limited by the resources that are available to the Content Manager OnDemand server.

For example, for disk and I/O capacity, each retrieve requires that the data is obtained from the archive (Tivoli Storage Manager, OAM, ASM, and cache). This data is on a disk or another storage device. The storage device retrieval rate is part of the total response time that is observed at the client, and both of them are affected by the following resources and system demand:

- ▶ Real memory: The data that is retrieved from disk must be stored in memory for it to be processed. Virtual memory allows large amounts of data to be swapped in and out of real memory, but it does not remove the need for real memory.
- ▶ Processing: Any data transformations that are performed on the Content Manager OnDemand server require available processing capability. If the capability is not available, the server waits until it becomes available. This wait lengthens the total response time of the client request.
- ▶ Concurrent retrievals: Each retrieval requires resources on the server. The higher the number of concurrent retrievals, the greater the amount of resources that are needed to complete the work in an acceptable amount of time.
- ▶ Network bandwidth: The retrieved data is sent to the clients over the Internet Protocol network. If the network bandwidth is not wide enough to satisfy all of the concurrent requests, the response time to the clients is slower and data is queued up in the server buffers, further slowing down the system.

13.3.3 Retrieval testing

The goal of retrieval testing is to verify that, under stressful system conditions, the maximum number of concurrent users can be served while at the same time the system meets the business requirements. The following process is a good general approach to retrieval testing of the system:

- ▶ Transaction type: Different types of transactions present different types of workloads on the system. For example, logon, document query, and document retrieval all use different components of the Content Manager OnDemand system. For each transaction type, measure the throughput and response time for a number of concurrent users that exceed the maximum predicted number. Tune and add resources to the system as needed until the system exceeds the service level agreement (SLA) requirements.
- ▶ Data types: The stored documents might be different sizes and data types (and might invoke preview exits). Multiple document retrieval tests must be run to verify the performance for the various types of stored documents.
- ▶ User workloads: The users that access the system might all exhibit the same usage patterns or might exhibit two or more usage patterns. The following process shows an example usage pattern:
 - a. Log on.
 - b. Wait five seconds.
 - c. Issue a document query with a maximum hit list size of 12 documents.
 - d. Wait five seconds.
 - e. Retrieve a 10 KB document.
 - f. Wait 40 seconds.
 - g. Retrieve a 20 KB document.
 - h. Wait 60 seconds.
 - i. Log off.

For example, a total of 50 concurrent users might follow this pattern. Also, other patterns might run at the same time. So, the user workload test must model this behavior and it must also be able to meet the business requirements at peak loads.

- ▶ Test driver location: The code that generates the retrieval workload can be installed on either of the following machines:
 - The same server on which the Content Manager OnDemand system is installed.
By using the same server, you can maximize the stress on the Content Manager OnDemand system by eliminating the network connection and by using system processing cycles to generate and measure the response time and throughput.
 - A network-connected workstation.
This situation simulates either a web server that connects to the Content Manager OnDemand server or a user that connects to the Content Manager OnDemand server.
- ▶ Number of test drivers: The number of systems that issue the requests can be increased so that the number of concurrent requests that reach the Content Manager OnDemand server exceeds the maximum expected number of requests.
- ▶ Test measurement: Two sets of measurements are used. The first set is at the test driver, which represents the user or Content Manager OnDemand Client. At this location, both throughput and response time on a transaction basis need to be collected. Also, it is important to check that the system that issues the retrieve requests is not overloaded and therefore not the performance bottleneck. In addition, at the Content Manager OnDemand server, request service times can be observed in the Content Manager OnDemand system log. System performance measurements must be collected by using operating system-specific tools.

13.3.4 System testing

After the load and retrieval tests are performed individually, it is important to perform an overall system test. This test must include running *everything* in parallel up to the maximum expected system usage. Everything includes load, retrieval, expiration, migration, duplication, and backup operations. The goal is to ensure that under the most stressful conditions possible, the system meets business requirements.

Note:

- ▶ The performance tuning process demands great skill, knowledge, and experience, and it cannot be performed by only analyzing statistics, graphs, and figures.
- ▶ The goal is to tune the Content Manager OnDemand server. You can “see” the bottlenecks in the server only if both the client and the network are clear of bottlenecks.

13.4 Performance issues that are based on data type

This section describes issues that relate to individual data types that can significantly affect the overall performance of Content Manager OnDemand. Several issues can be addressed by selecting or clearing certain functions and features within Content Manager OnDemand. Several of the issues that we describe can be addressed only by changing the way in which the data is produced from the source.

13.4.1 PDF data

Portable Document Format (PDF) data is an increasingly common data type that can be archived within Content Manager OnDemand. The following key advantages are available by using this data type as a document format:

- ▶ It is a read-only format that does not require any external resources, such as images or fonts. It is self-contained.
- ▶ The viewer for PDF can be downloaded at no charge from the Adobe website and the browser plug-ins for PDF are also available at no charge.

During PDF document creation, resources, such as images and custom fonts, are placed in the data stream once and then referenced many times from within the PDF file. If a large report is produced from many small documents, that report requires only one copy of the resources.

However, when the PDF is indexed, the PDF Indexer creates many PDF documents from the input file. Each of these documents requires a certain number of PDF structures, which define a document. These documents are concatenated together in the .out file, and then loaded into Content Manager OnDemand as separate documents. Because the resources are extracted and placed into a separate resource file, they are not included in each document. For an illustration of the process, see Figure 13-3.

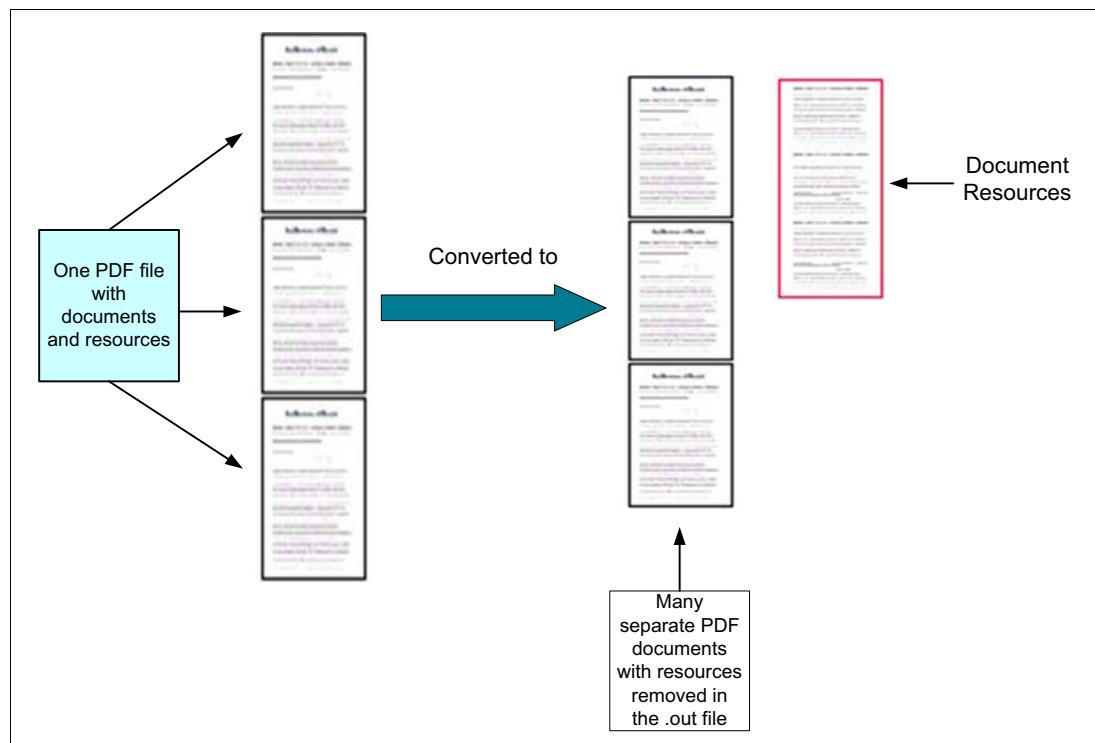


Figure 13-3 PDF indexing

If no resources are collected, the size of the .out file, which contains all of the individual documents, might be larger than the original file. For tips about how to reduce the size of the output file, see 7.3.5, “PDF indexing: Using internal indexes (Page Piece Dictionary)” on page 173.

The size of the input file and the output file can create problems during the load process:

- ▶ The temporary space that is used during indexing can be too small and the load fails.
- ▶ The maximum input file size that the PDF Indexer can process is 4 GB, but the recommended maximum size for a single document (after indexing) is 50 MB. If this size is exceeded, the system might run out of disk space or memory.

Create PDF data with the base 14 fonts, which do not need to be included in the PDF file. Because they are not included in the PDF file, they are not extracted during resource collection, which improves performance. For more information about the PDF data stream and fonts, see 7.3.1, “PDF fonts and output file size” on page 166.

13.4.2 Line data

Line data (ASCII or EBCDIC text-based reports) is the most common type of data that is stored in Content Manager OnDemand. The type of line data that we describe here is a special form of transaction-style report, where it is necessary to search on a value that appears on every line of the report. This transaction data has a transaction number that appears on every line and must be sorted either by column or row and either ascending or descending.

When you index transaction data, if each transaction number from each line of the report is treated as a database index, such as date or customer name, the database becomes large quickly. Content Manager OnDemand has a special type of field for transaction data, which is illustrated in Figure 13-4 by the boxed data on the left of the window.

CENTRAL ADJUSTMENTS DEPARTMENT								
INCOMING SEQ NO.	OUTGOING ID	KB SEQ	ENDPOINT ID	DT	ITEM AMOUNT	CASH AMOUNT	LETTER AMOUNT	ROUTING TRANSIT
P000000072	21593.34	1	0000-0032	TR	50.00	21593.34	0000-0371	
P000000073	2151.39	1	0000-0194	TR	50.00	2151.39	0000-0040	
P000000074	2151.39	2	0000-0194	TR	20.00	2151.39	0000-0040	
P000000075	2151.39	3	0000-0194	TR	10.00	2151.39	0000-0040	
P000000076	2151.39	4	0000-0194	TR	40.00	2151.39	0000-0040	
P000000077	2151.39	5	0000-0194	TR	296.00	2151.39	0000-0040	
P000000078	2151.39	6	0000-0194	TR	77.33	2151.39	0000-0040	
P000000080	2151.39	7	0000-0194	TR	127.00	2151.39	0000-0040	
P000000081	2151.39	8	0000-0194	TR	25.00	2151.39	0000-0040	
P000000082	2151.39	9	0000-0194	TR	135.00	2151.39	0000-0040	
P000000084	2151.39	10	0000-0194	TR	300.00	2151.39	0000-0040	
P000000085	2151.39	11	0000-0194	TR	25.00	2151.39	0000-0040	
P000000086	2151.39	12	0000-0194	TR	11.00	2151.39	0000-0040	
P000000089	2151.39	13	0000-0194	TR	206.00	2151.39	0000-0040	
P000000091	8175.12	1	0000-0372	TR	264.75	8175.12	0000-7083	
P000000093	2151.39	14	0000-0194	TR	233.00	2151.39	0000-0040	
P000000094	2151.39	15	0000-0194	TR	96.90	2151.39	0000-0040	
P000000095	1802.24	1	0000-0502	TR	638.00	1802.24	0000-1544	
P000000097	21593.34	2	0000-0032	TR	341.54	21593.34	0000-0589	

Figure 13-4 Transaction data in graphical indexer

The transaction data field selects the *first* and *last* values from a group of pages and only these group level values are inserted into the database. Content Manager OnDemand queries the database by comparing the search value that is entered by the user to two database fields, the beginning value and the ending value. If the value that is entered by the user falls within the range of both database fields, Content Manager OnDemand adds the item to the document list.

From a performance perspective, the use of the transaction data field for transaction-style line data optimizes indexing performance by reducing the number of index values to be inserted into the database. Therefore, the process of loading and retrieving these large reports is faster and the Content Manager OnDemand database is many times smaller.

13.4.3 AFP data

AFP data is a multi-part data type. In addition to the variable data, external resources, such as images, fonts, and logos, are also referenced by the AFP data stream. When Content Manager OnDemand stores AFP data, the resources are also archived. When the data is viewed, the referenced resources are displayed.

It is a common misconception that if fonts are collected when the data is loaded, they are available for viewing in the Windows client. However, Windows does not recognize AFP fonts. It is not possible to use these fonts even if they are sent to the client as part of the resource. Windows clients require a mapping from AFP fonts to Adobe Type Manager (ATM) fonts or TrueType (TT) fonts. Content Manager OnDemand provides this mapping for most standard fonts. For more information about mapping custom fonts, see *IBM Content Manager - Windows Client Customization Guide and Reference*, SC27-0837.

One possibly useful implementation of storing fonts with the resource group is when server reprint is necessary. If the fonts are stored with the resource group, they can be retrieved from Content Manager OnDemand and used by AFP printers. However, if fonts are collected, they are also sent to the client as part of the resources group and then discarded. Storing the fonts with the resource group serves only to increase network traffic when transferring the resource to the workstation. A more practical option for server printing is to store the font in a fontlib and to keep only the reference (path) to the fontlib. Although the font is accessible on the server, Print Services Facility (PSF) or InfoPrint does not need the font to be inline (stored in the resource group). The use of this approach also allows all AFP data that references the font to use the single instance of the font without redundant inline storage.

Figure 13-5 on page 311 shows the indexer information in the application where you can select the resources to collect with the Restype= parameter. Unless reprints to AFP printers with 100% fidelity is a requirement, do not collect the fonts.

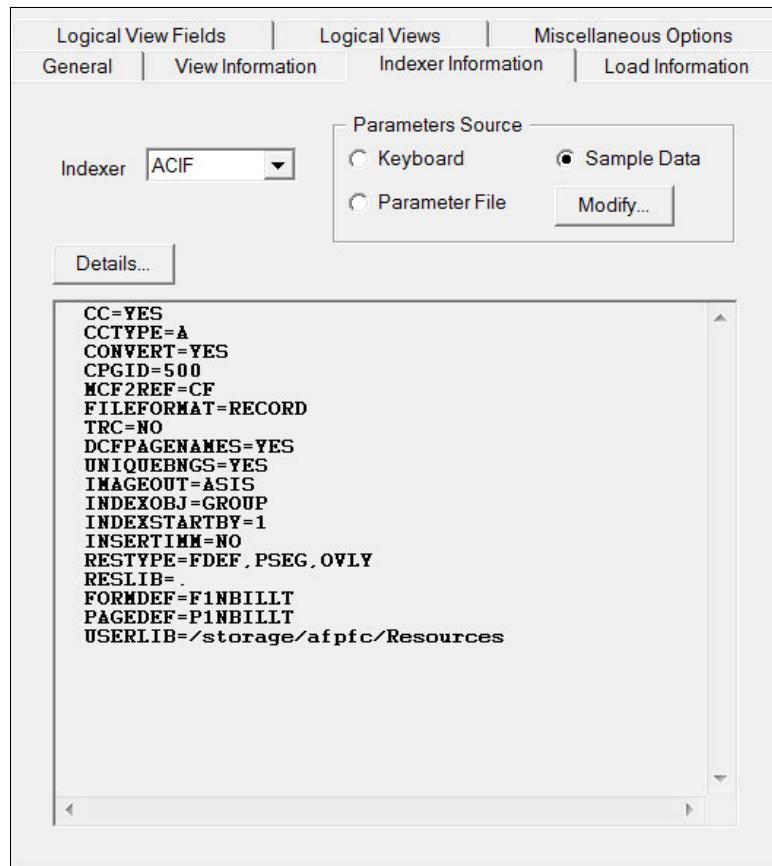


Figure 13-5 Collecting AFP fonts

The Content Manager OnDemand for i server does not collect the fonts and it does not give the administrator that option. The Resource Information window (under Indexer Properties) is not available to the Content Manager OnDemand for i administrator. If you are reprinting to an AFP printer, the fonts must be available on the IBM i server, or font substitution is performed.

13.4.4 Image data

To optimize performance with storing and retrieving image formats, such as Tagged Image File Format (TIFF), Graphics Interchange Format (GIF), and Joint Photographic Experts Group (JPEG), do not compress the data because the file sizes might increase. To turn off compression, select the **Disable** option from the Load Information tab within the application. See Figure 13-6.

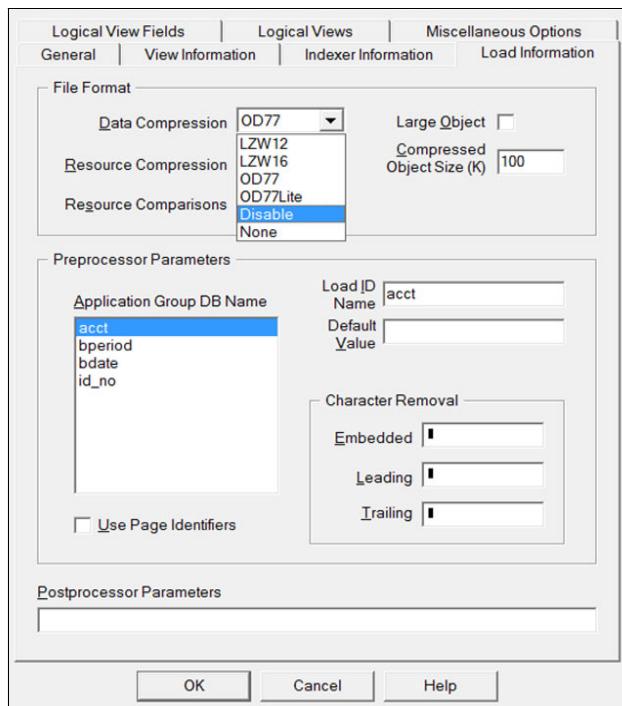


Figure 13-6 Disabling compression

Two methods are available to turn off data compression:

- ▶ **Disable:** Content Manager OnDemand does not compress the input data. Choose this option when the input data, such as PDF and compressed TIFFs, is already compressed. Documents are extracted by the appropriate viewer on the client (for example, Adobe Acrobat Reader).
- ▶ **None:** Content Manager OnDemand does not compress the input data when it loads the input data into the system. When the user selects a document for viewing, Content Manager OnDemand compresses the document before it transmits it over the network and extracts the document at the client.



Part 4

Enhancement options

This part contains the following chapters:

- ▶ Chapter 14, “Report distribution” on page 315
- ▶ Chapter 15, “Full text search” on page 335
- ▶ Chapter 16, “Enhanced Retention Management” on page 353
- ▶ Chapter 17, “Content Federation Services for Content Manager OnDemand and IBM Enterprise Records” on page 365



Report distribution

IBM Content Manager OnDemand Distribution Facility (ODF) is an optional report distribution feature for IBM Content Manager OnDemand. ODF provides an easy way to automatically group reports and portions of reports and distribute the reports to multiple users. ODF distributions can be printed, created as an output file, or emailed as an attachment.

ODF can distribute reports that are stored in a Content Manager OnDemand server on any platform that is supported by Content Manager OnDemand.

In this chapter, we cover the following topics:

- ▶ Introduction to Content Manager OnDemand Distribution Facility
- ▶ Defining the objects with the Administrator Client
- ▶ Defining the objects by using batch administration
- ▶ Customizable user exits
- ▶ Status and monitor tool

14.1 Introduction to Content Manager OnDemand Distribution Facility

Before Content Manager OnDemand version 9.5, two report distribution components were available:

- ▶ OnDemand Distribution Facility for z/OS
- ▶ Report Distribution Facility for Content Manager OnDemand for Multiplatforms

Both of these components contained certain strengths and weaknesses. In V9.5, the strengths of both of these components were merged into a single component named OnDemand Distribution Facility (ODF), which offered the following advantages:

- ▶ It runs on all Content Manager OnDemand platforms.
- ▶ It can run on a separate platform from where the Content Manager OnDemand server is installed.
- ▶ Its operation can be monitored through a new graphical monitor, the OnDemand Monitor.
- ▶ It includes transform support where Content Manager OnDemand can transform content from one data type to another data type before the content is sent as part of an ODF distribution.

This chapter describes ODF V9.5. For any new installations (on z/OS or AIX) before version 9.5 of Content Manager OnDemand, we suggest that you install ODF.

Figure 14-1 shows the evolution and merger of ODF 9.5 from its predecessors ODF9.0 and Report Distribution System (RDF) 9.0.

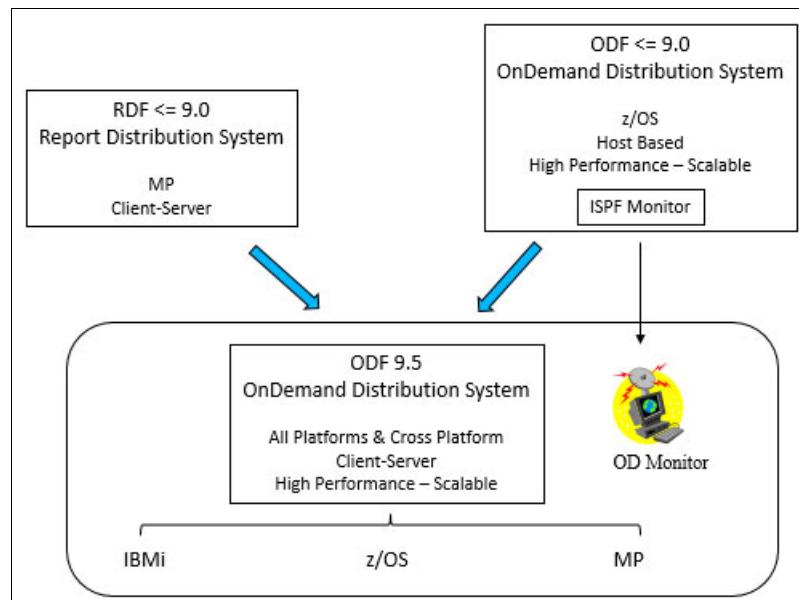


Figure 14-1 Evolution of ODF

When you load documents into Content Manager OnDemand, you might need to print these documents or send them to various people in your organization.

Content Manager OnDemand automates the process of sending the documents that are loaded into Content Manager OnDemand to print (or the JES spool), a file (or a z/OS dataset), to a recipient as an email attachment, or to a recipient as an email notification.

Another benefit to using ODF is that you can select and combine documents from different reports and organize them by defining their order and separating them by using banner pages.

Figure 14-2 is an overview of the OnDemand Distribution Facility and its interaction with the Content Manager OnDemand server.

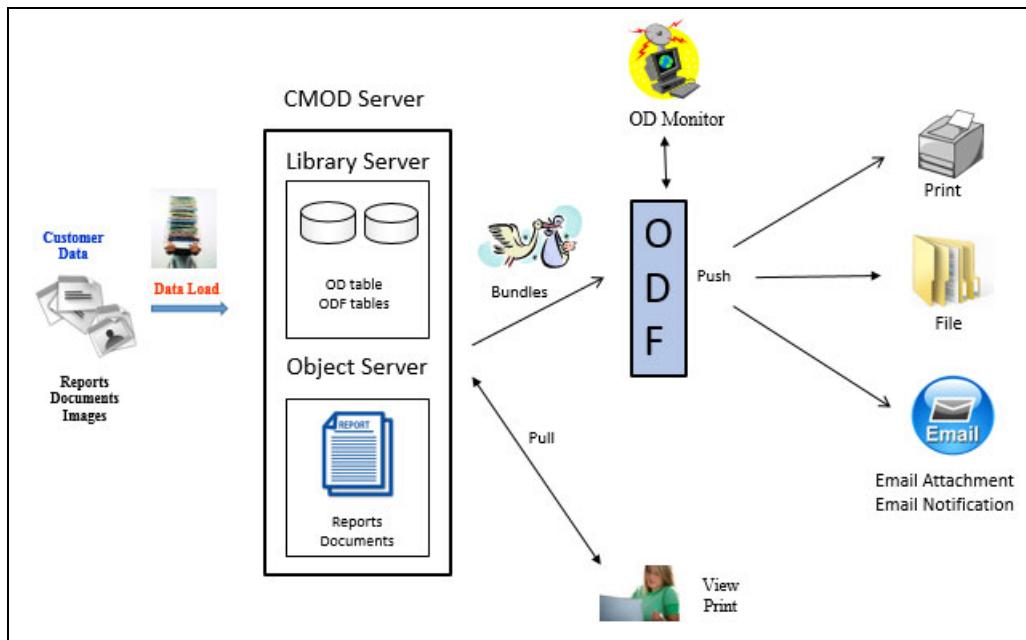


Figure 14-2 Content Manager OnDemand Distribution Facility overview

Figure 14-2 shows that the Content Manager OnDemand server and its operation did not change. Reports and documents are loaded into the server, and system users continue to view and print their documents normally. The only addition to the library server is a set of ODF tables that define the documents that are to be distributed to which users and when. The ODF process reads the ODF tables and collects the required documents and bundles them for each recipient. ODF then send out the “bundles” to the appropriate destinations (email, file, and print). Alternatively, ODF can send each recipient (based on system definitions) an email notification that the report and document were loaded and are available for viewing.

Different organizations have different report and document load and retrieval patterns. In certain cases, documents are loaded and never retrieved. In other cases, a loaded document is retrieved multiple times by multiple users. In other cases, it is known that when a specific report or document is loaded, one or more copies must be distributed to one or more destinations. What benefit does automating this distribution process provide?

The biggest benefit is that as reports are loaded into Content Manager OnDemand regularly, they can be delivered automatically to one or more users as they are loaded. Also, after the distribution is set up, no other changes are required, such as changing the document selection criteria to identify the latest data that is loaded.

For example, suppose that your organization generates monthly statements for your customers. You must store these documents in Content Manager OnDemand, and you must print the statements and mail them to the customers. With ODF, you can set up a distribution that automatically retrieves these documents as they are loaded into Content Manager OnDemand and sends them to a spool file for printing.

Another example is a sales team that generates a monthly sales report for each person on the sales team. The sales manager needs a copy of these reports. A distribution can be set up to email the documents to the appropriate sales manager.

The applications for using ODF are endless, but the basis for using it is the same. Documents are loaded regularly and are needed by one or more users as they become available in Content Manager OnDemand. Let us look at a specific example from our fictitious company that was introduced in 1.2.1, “Background information of an example company” on page 6.

AFinancial Co generates monthly credit card statements for all its customers. These customers can choose to receive a hardcopy of the statement or have the statement sent to them as an email attachment.

In this example, even though separate customer statements are created each month, they are loaded into the system at the same time, so only one load occurs each month. This information is important when you are determining the best way to set up the distribution. Before a distribution is set up, ask yourself the following questions:

- ▶ What documents are needed?
- ▶ Who receives the documents?
- ▶ When are the documents retrieved and delivered?
- ▶ Where are they delivered?

14.1.1 What documents are needed

In our example, we identified our documents as the customer statements. How do you identify the customer report that you need from the hundreds of thousands of documents that are stored in Content Manager OnDemand? Certain customers might receive multiple monthly statements.

In general, you identify the documents by creating an SQL query that uses index fields and values that uniquely identify the documents that you want to retrieve when they are loaded. You can then define the distribution to include multiple report bundles with different SQL queries for each bundle. If the SQL must retrieve the document that is the same except for a value that identifies the recipient, a single distribution can be used with a recipient list. In this case, the SQL specifies a wildcard value. When processing, ODF fills in the recipient ID in the SQL statement. For example, a recipient list contains recipients 100001, 100002, and 100003 and an SQL statement of “`Where branch_id = '$ODF_RECIPIENT'`”. When this recipient list is processed, ODF creates a distribution for recipient 100001 with all reports where `branch_id = '100001'`, recipient 100002 will receive a distribution that contains all reports where `branch_id = '100002'`, and so on.

14.1.2 Who receives the documents

In our example, each customer needs a statement copy every month. To identify the customers to Content Manager OnDemand, an ODF recipient must be created for each customer. Depending on how the documents are delivered, a destination must be set up. For example, if a set of documents will be delivered to a recipient by using email, an email address must be specified in the recipient definition.

14.1.3 When the documents are retrieved and delivered

ODF operates throughout the 24-hour day. You can schedule your distributions to be processed at a specific time of day or processed as they are loaded. To specify when the distribution is delivered, choose the method, which is either Loaded, All Ready, Time of Day, Time of Print, or external.

ODF operates on a 24-hour clock: 00:00 to 23:59. If a time of distribution (TOD) of 01:00 is specified and documents are loaded at 23:30, the documents are processed immediately and they do not wait until the next day because the TOD specified was reached for that 24-hour day.

14.1.4 Where are they delivered

You can deliver the distribution to a printer (or the JES spool on z/OS) for printing, a file (or TSO dataset on z/OS), or an email attachment. Alternatively, you can specify that the documents will not be distributed at all and that an email notification that the documents were loaded is sent to the specified recipients. In our example, certain customers specified that they want their statements to be delivered by email, and other customers specified that they want a hardcopy.

14.1.5 Cross-platform access

ODF (running on any supported platform) can access a Content Manager OnDemand instance that is running on any (local or remote) platform that is supported by Content Manager OnDemand. For more information about how to configure ODF, see “Configuring ODF” in the *OnDemand Distribution Facility Installation and Reference Guide*, SC19-3358.

14.2 Defining the objects with the Administrator Client

After you set up the Content Manager OnDemand (application group and application) objects, you are ready to set up the ODF objects. This section describes the definition of the ODF objects by using the Content Manager OnDemand Administrator Client (Figure 14-3).

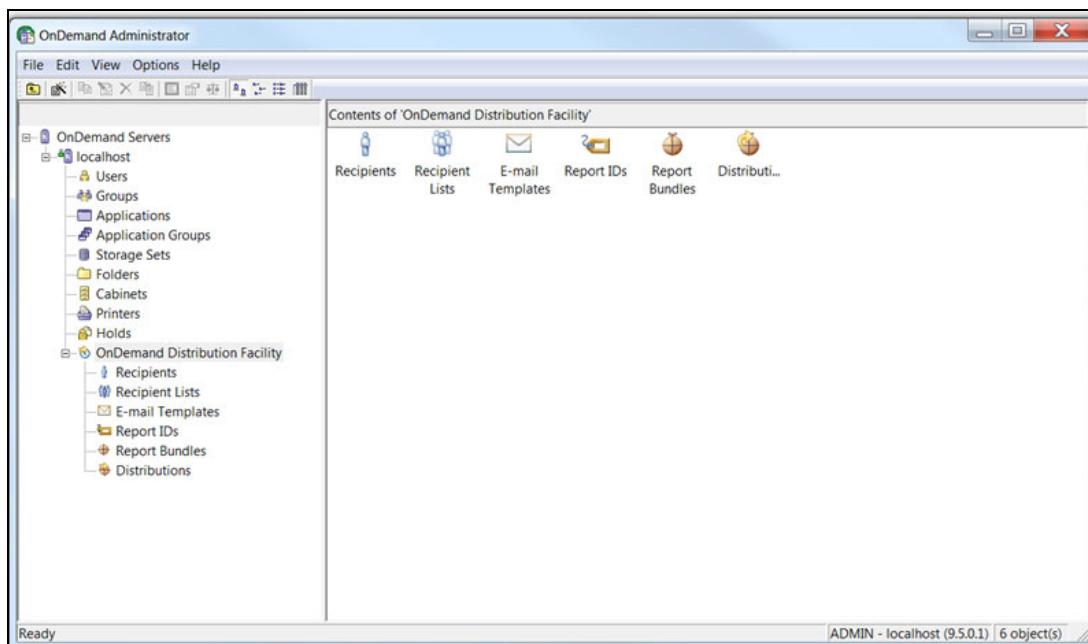


Figure 14-3 Administrator Client ODF objects

14.2.1 Adding a recipient

The recipient object contains all of the information about the recipient of the distribution. The only required field is the recipient ID, which, when combined with the distribution name, uniquely identifies the distribution.

Figure 14-4 shows the window where you add a recipient.

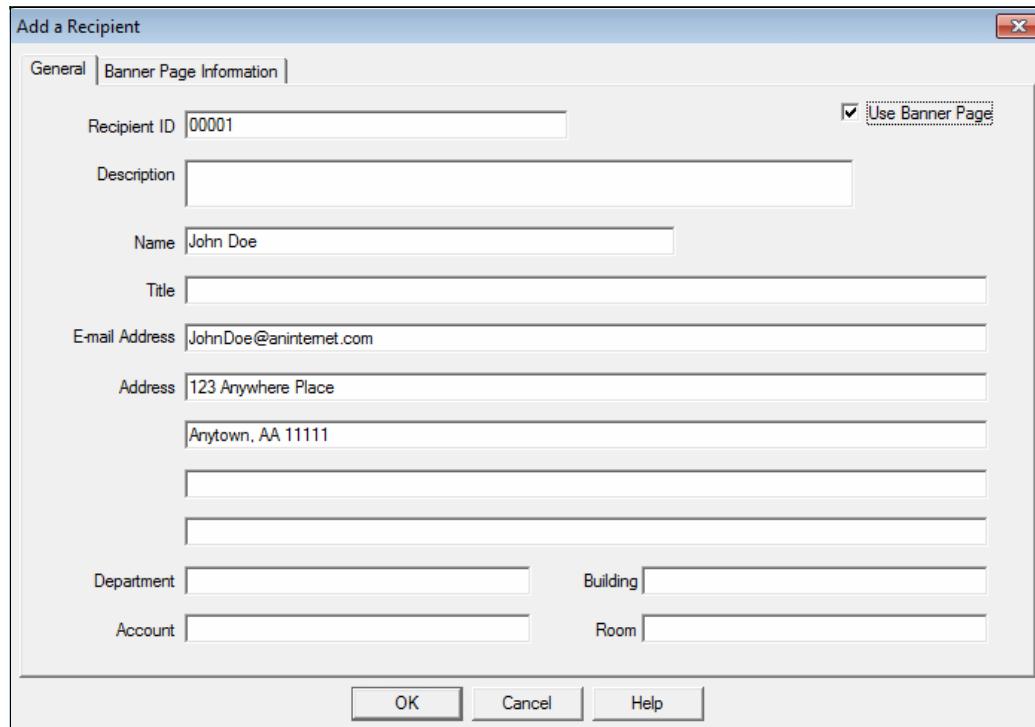


Figure 14-4 Add a Recipient

Recipients who receive a printed copy of the distribution can choose to include a banner page in the distribution by selecting **Use Banner Page**. You can specify up to eight header lines to include in the banner page, as shown in Figure 14-5 on page 321.

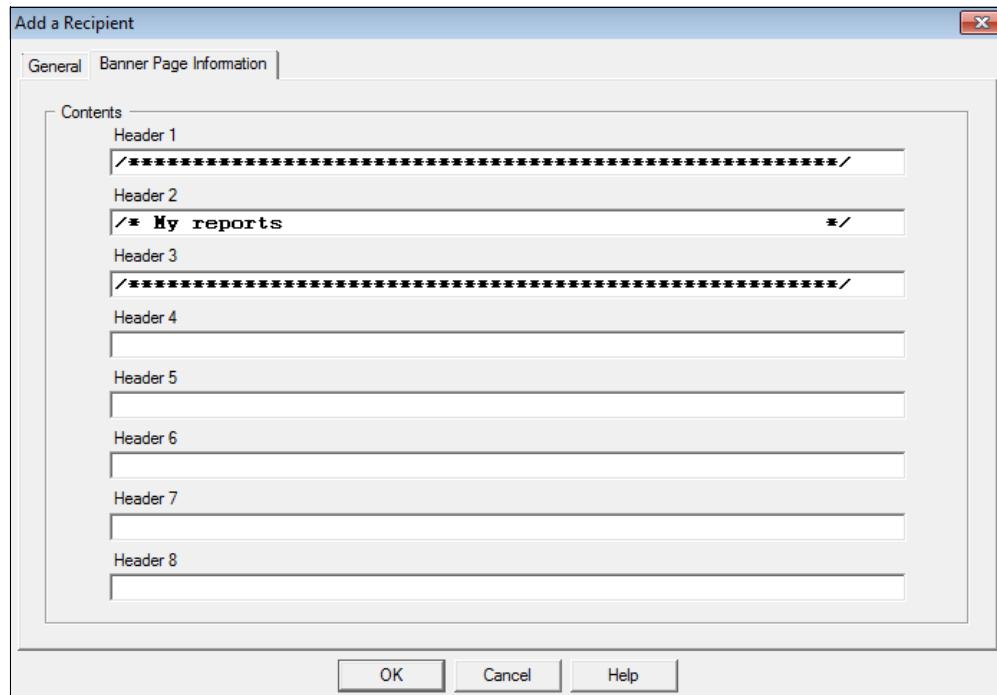


Figure 14-5 Specifying banner page information

14.2.2 Adding a recipient list

If several recipients must receive the same reports at the same time, you can create a recipient list. With this list, you create a single distribution that is sent to every recipient in the list.

Recipients are added to the list by selecting the ID on the left and clicking **Add**, as shown in Figure 14-6 on page 322.

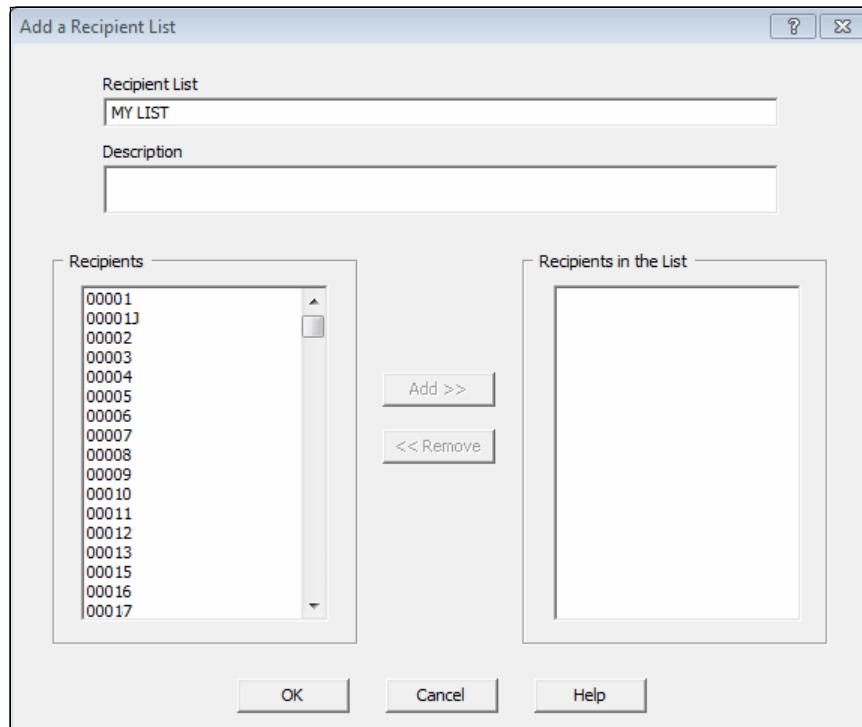


Figure 14-6 Adding a recipient list

14.2.3 Adding a report ID

The next step is to define the reports to ODF. The report ID identifies the application group and application to which the report belongs. Figure 14-7 shows the window where you add the report ID.

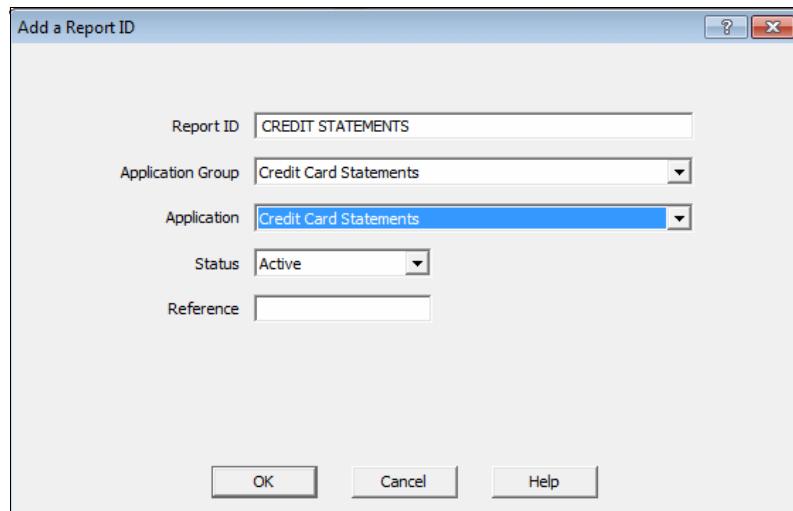


Figure 14-7 Adding a report ID

To create a report ID, specify the identifier and then choose the application group and application from the drop-down selection.

Use the reference field to control when a report is available for distribution. The value that you enter in this field is used with a marked index column in the Content Manager OnDemand application group. When a reference value is available, the reference value is matched with the index value of the report. If a match exists, the report is made available for distribution. This tool is useful if several drafts of a report exist and you want to distribute only the final version.

14.2.4 Adding a distribution

Now that the recipients and report IDs are set up, it is time to create the distributions. In the distribution definition, you specify when, where, and how the distribution is delivered. In our example, we create a distribution that is processed while the documents are loaded. The output is sent as an email. For a sample distribution definition, see Figure 14-8.

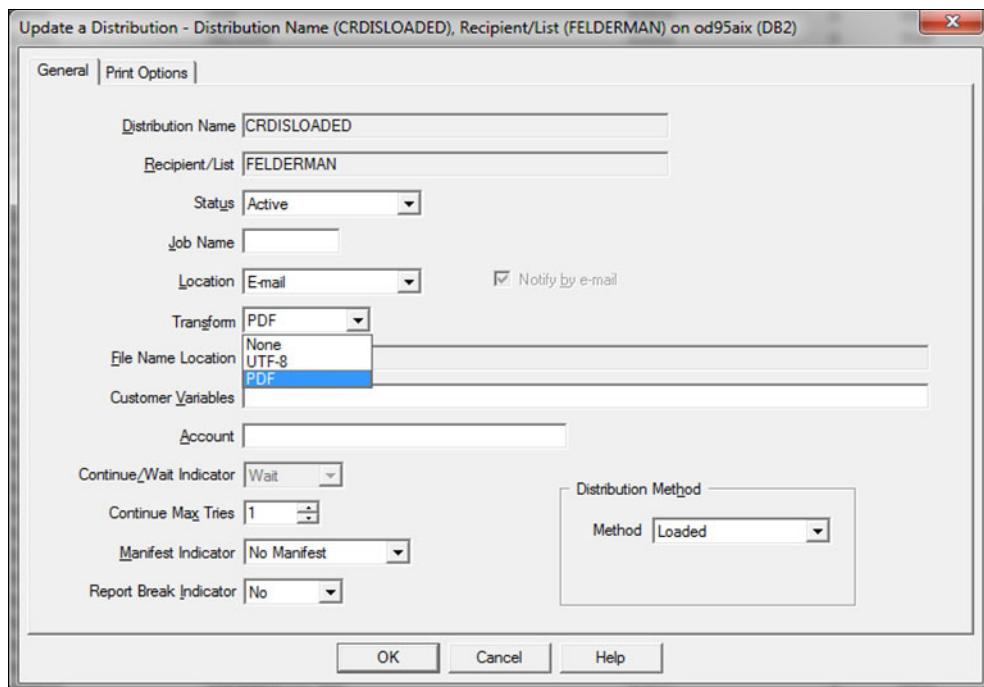


Figure 14-8 Adding a distribution

Distribution Name

With the recipient or recipient list name, the distribution name uniquely identifies the distribution. For our example, we name this distribution CREDIT CARD STATEMENTS.

Recipient/List

Choose your recipient. For our example, we add the newly created recipient from the drop-down menu.

Status

Two values are valid for status:

- ▶ Active indicates that the distribution is processed while the documents are loaded.
- ▶ Inactive indicates that the distribution is *not* processed while the documents are loaded.

Job Name

To improve ODF performance, you can use a submitted job and the persistence feature. When you use a job name on distributions, ODF uses a feature of z/OS that allows jobs to run in created address spaces. The ODF distribution runs under the job name that is specified. For our example, we leave the job name value blank.

Location

The location specifies where the distribution is delivered. We select **E-mail** for our distribution.

The following options are available for the Location field:

- ▶ **Print:** The output is sent to a JES spool file.
- ▶ **File:** The output is sent to a generation data set (GDS) if a dataset value is specified. Otherwise, it is sent to a TSO dataset.
- ▶ **None** (with “Notify by e-mail” selected): An email is sent to the recipient to notify the recipient that the distribution is available.
- ▶ **Email:** The output is sent as an attachment in an email to the recipient.

Note: The “Notify by e-mail” check box is available for use with Location values of Print, File, or None. The selection of the “Notify by e-mail” check box sends an email to the recipient to notify them that the distribution is available.

Customer Variables

This field contains any information that you might need to pass to the customizable user exits. For example, if this distribution requires special spool file allocation options, you can enter the information in this field. The preallocation exit can then use the information to change the spool file allocation parameters. For our example, we leave this field blank.

Account

This field is optional. This field specifies the name to use on the JCL job card. For our example, we leave this field blank.

Distribution Method

The distribution method controls the scheduling and processing of the distribution. Because we want the distribution to be processed while the documents are loaded, we select the **Loaded** method.

The following distribution methods are available:

- ▶ **Loaded:** The distribution is scheduled for processing when the first report bundle is archived or stored in Content Manager OnDemand. The distribution is submitted for print processing when all of the report bundles in the distribution with a Wait/Ignore Indicator of Wait are loaded.
- ▶ **All Ready:** The distribution is scheduled for processing when the ODF address space is started. The distribution is submitted for print processing when all of the report bundles in the distribution with a Wait/Ignore Indicator of Wait are loaded.
- ▶ **External:** A process outside of ODF schedules the distribution. The distribution is submitted for print processing when all report bundles that are defined with a Wait/Ignore Indicator of Wait are loaded.

- ▶ Time of Print: The distribution is scheduled when the first report bundle of the distribution is archived or stored in Content Manager OnDemand. Before the Time of Print time, the distribution is submitted for print processing whenever all of the report bundles with a Wait/Ignore Indicator of Wait are loaded. If all of the required reports are not available at the specified time, when the Time of Print time is reached, the distribution is submitted for print processing with whatever report bundles are available.
- ▶ Time of Day: The distribution is scheduled at the specified time of day. It is submitted for print processing when all of the report bundles defined with a Wait/Ignore Indicator of Wait are loaded.
- ▶ Time: The time when the distribution is processed. The default value is the current time. This field displays only if the distribution method is set to Time of Day, or Time of Print.

Continue/Wait indicator

This option is valid only when the Distribution Method is Time of Day or Time of Print. From the drop-down list, select either **Continue** to continue processing report bundles as they are available after the Time is reached, or select **Wait** to wait until the next Time occurrence to print any report bundles that become available.

Continue Max Tries

This value controls the maximum number of attempts that are made to process the report bundles.

Manifest Indicator

This value indicates whether a manifest page, which lists the report bundles that are included in the distribution, must be created. The manifest defaults to a separate file. If you want the manifest in the same file as the report bundles, specify **Manifest in Sysout**.

Report Break Indicator

This value indicates whether the report bundles must be included in the same file or broken up into separate files.

Print Options tab

Use the Print Options tab to specify the allocation values to use for the JES spool file. These options do not apply to our example distribution.

14.2.5 Adding a report bundle

After you define the distribution, you must define the report bundles that are included in the distribution. To add a report bundle, right-click the distribution that you added and then select **Add Report Bundle**. See Figure 14-9 on page 326.

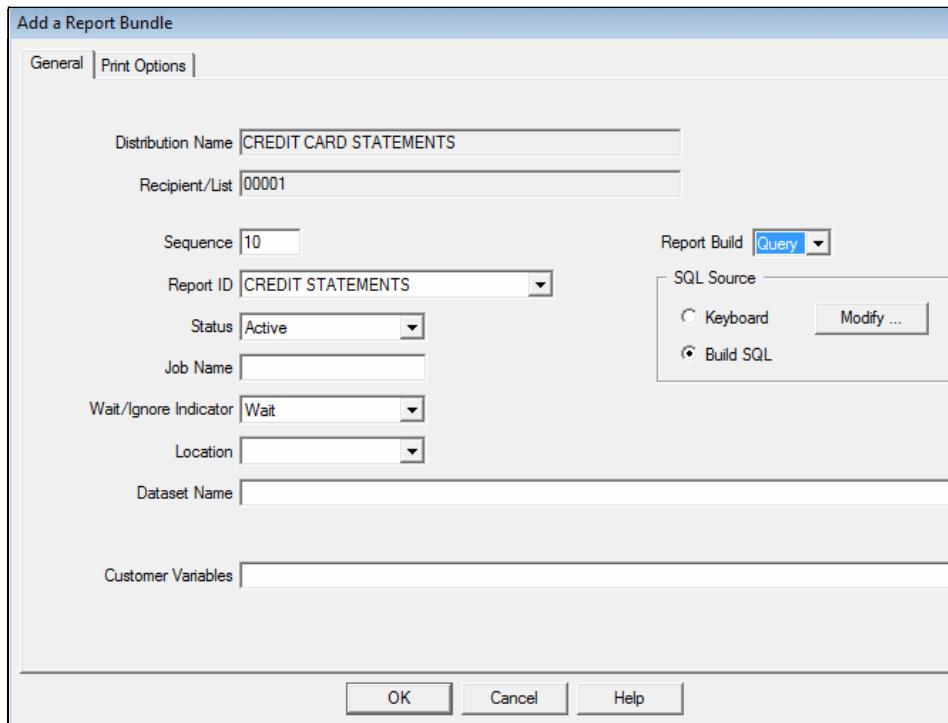


Figure 14-9 Adding a report bundle

Distribution Name and Recipient/Recipient List

The report bundle is created as a child object of the distribution. The values are not modifiable and disabled in the window.

Sequence

This value identifies the sequence that the report bundles are included in the distribution. The default is 10, and each new report bundle increments the sequence by 10. This value provides flexibility to add report bundles without the need to renumber any other report bundles.

Report ID

This number identifies the report to include. For our example, we select the previously added report ID from the drop-down menu.

Wait/Ignore Indicator

When more than one report bundle is specified in the distribution, this value tells ODF whether this report bundle must be available before the distribution is processed. A value of Wait indicates that you wait until this report is loaded before the distribution processing begins. A value of Ignore indicates that the distribution is processed even if this report bundle is not available. This function is useful if documents are loaded at different times but you want them to be processed and included in a single distribution instance.

Report Build

This field indicates whether the distribution must include the full report or if a query will be performed and only a portion of the report will be included. When Query is selected, the SQL source option is available to build the query. You can either type the query by using the Keyboard option or build the SQL, as shown in Figure 14-10 on page 327. For our example, we build a query to include only the statements for John Doe.

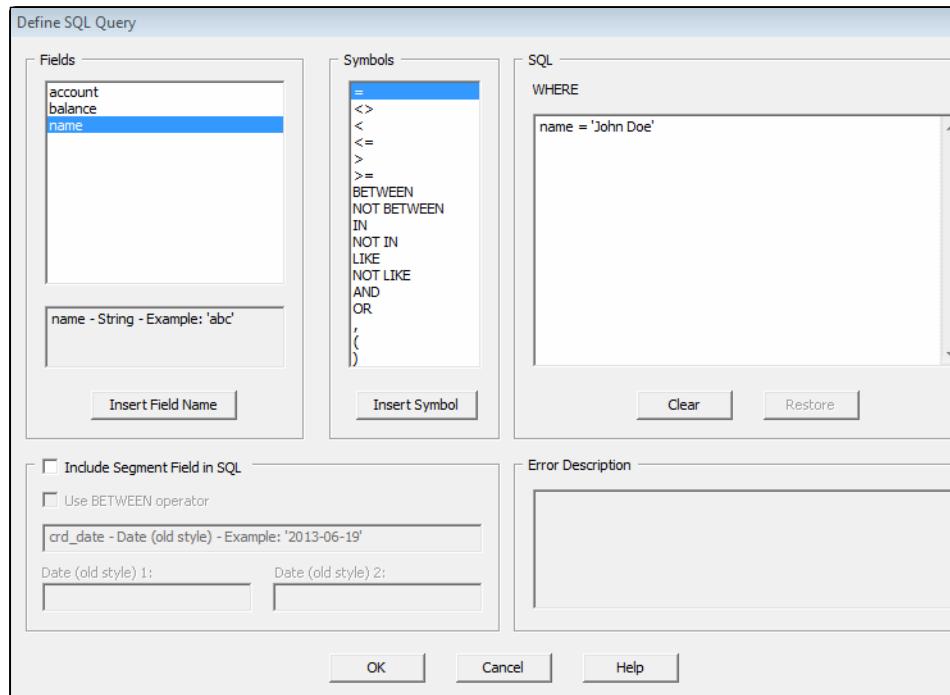


Figure 14-10 Building the SQL query

Additionally, users can specify a wildcard with a substring in the SQL statement. On execution, ODF will substitute the correct portion of the recipient or recipient list name.

The format of the wildcard is shown:

- ▶ \$ODF_RECIPIENT(*start pos:length*) where *start pos* is the number of the characters to start and *length* is the number of characters to use. (*start pos:length*) is optional.
- ▶ \$ODF_RECIPLIST(*start pos:length*) where *start pos* is the number of the characters to start and *length* is the number of characters to use. (*start pos:length*) is optional.

Job Name, Location, Dataset Name, and Print Options

These fields can be used to override the values that are specified in the distribution definition. Use this capability to specify the values at the distribution level that apply to most of your report bundles and still customize for individual report bundles.

14.3 Defining the objects by using batch administration

ARSXML provides a batch interface to add, update, delete, or export a list of ODF objects. We show the `arsxml` command and a sample XML file that is used to create each of the objects that we added earlier.

14.3.1 Recipient

Run the following command to add a recipient:

```
Arsxml add -h myod -u myuser -p mypwd -v -i /recipientAdd.xml
```

Example 14-1 on page 328 shows the content of our example `recipientAdd.xml` file.

Example 14-1 recipientAdd.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ondemand.xsd">
<odfRecipient name="00001"
    fullName="John Doe"
    email="JohnDoe@aninternet.com"
    addr1="123 Anywhere Place"
    addr2="Anytown, AA 11111"
    banner="true"
    header1="/*****"
    header2="/*           */"
    header3="/*****"  />
</onDemand>
```

14.3.2 Report ID

Run the following command to add a report ID:

```
Arsxml add -h myod -u myuser -p mypwd -v -i /reportIDAdd.xml
```

Example 14-2 shows the content of our example reportIDAdd.xml file.

Example 14-2 reportIDAdd.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ondemand.xsd">
<odfReportId name="CREDIT STATEMENTS"
    status="Active"
    applicationGroup="Credit Card Statements"
    application="Credit Card Statements"  />
</onDemand>
```

14.3.3 Distribution and report bundle

Run the following command to create a distribution and report bundle:

```
Arsxml add -h myod -u myuser -p mypwd -v -i /distributionAdd.xml
```

Example 14-3 shows the content of our example distributionAdd.xml file.

Example 14-3 distributionAdd.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<onDemand xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ondemand.xsd">
<odfDistribution name="CREDIT CARD STATEMENTS"
    recipient="00001" s
    status="Active"
    location="E-mail document"
    manifest="Manifest"
    reportBreak="false"
    distMethod="Loaded"  >
<odfReportBundle sequence="10"
    reportId="CREDIT STATEMENTS"
```

```
        reportBuild="Query" >
    <sql>WHERE name = 'John Doe' </sql>
</odfReportBundle>
</odfDistribution>
</onDemand>
```

14.4 Customizable user exits

ODF provides several user exits with which you can tailor the system to meet your installation's requirements. You can optionally use the sample exits that are provided or customize the exit to meet your specific needs.

14.4.1 **arsodfxa: Spool file dataset allocation attributes exit**

You can use the **arsodfxa** spool file dataset preallocation exit to modify the currently defined ODF JES spool file dataset output parameter definitions that are used for dynamic allocation of the report and manifest JES spool file datasets. The **arsodfxa** exit is called when ODF detects a non-blank Customer Variables field in either the ODF distribution or report bundle definition, but only if the field value is not set to DO NOT SCHED or NOSCHED.

The output parameters that are specified in the report bundle definition and the output parameter string are passed to the **arsodfxa** exit. The exit can modify the output parameter string. The string that is returned from the user exit is used to allocate the JES spool file datasets for the report bundle and manifest JES spool file datasets.

14.4.2 **arsodfxb: Banner, header, and trailer exit**

On z/OS, use the **arsodfxb** exit to customize the banner information that is written to the JES spool file datasets. Banner information is written to the JES spool file dataset when the recipient definition requests that a banner is printed and the location of the report bundle is Print. ODF calls the **arsodfxb** exit for three different types of banner data:

- ▶ **Banner page**

Information to be written before the first report bundle in the distribution is written to the JES spool file dataset. The exit is called at the start of processing the first report bundle within the distribution with ODFBANNER-REQUEST-TYPE = 1 to process banner information.

- ▶ **Header page**

Information to be written before the second and each subsequent report bundle in the distribution. The exit is called before each subsequent report bundle within the distribution with ODFBANNER-REQUEST-TYPE = 2 to process the header information. See Example 14-4 on page 330.

- ▶ **Trailer page**

Information to be written to the JES spool file dataset after the report bundle is written. The exit is called after each report bundle is processed with ODFBANNER-REQUEST-TYPE = 3. The exit is passed information about the report bundle and recipient, and the exit uses this information to format the lines to display.

The exit returns a buffer of data. The maximum size is 10,240 bytes. The exit formats the data and adds a new line character x'15' wherever the data must start on a new line in the spool file.

Example 14-4 Banner header page sample output

```
*****  
/* My Reports */  
*****  
  
*****      *****      *****      *****      *****  
**   **  **  *  **          **   **  **  
**   **  **  **  *****      *****  *****  
**   **  **  **  **          **   **          **  
*****  *****  **          *****  *****  
  
*****BANNER PAGE*****  
  
*****  
*          *  
*  REPORT INFORMATION *  
*          *  
*  REPORT ID: CREDIT STATEMENTS *  
*  REPORT:          *  
*  REPORT DATE: 2013-06-20          *  
*          *  
*  PRODUCED FOR:          *  
*          *  
*  RECIPIENT: 0001          *  
*  REQUEST DATE: 2013-06-20          *  
*  REQUEST TIME: 07:46:55          *  
*          *  
*****
```

14.4.3 **arsodfxm: Bundle manifest exit**

On z/OS, the sample **arsodfxm** user exit is a COBOL program that enables you to customize the manifest output. The manifest consists of Header and Detail lines. ODF calls the bundle manifest exit with two functions: one function to process the header section of the manifest and the other function to process the detail lines.

When the request type is Header, the exit returns a buffer of data. The maximum size is 1,024 bytes. The exit formats the data and adds a new line character of x'15' wherever the data needs to start on a new line in the spool file.

14.4.4 **ODFProcessDist.java: Processed distribution exit**

Use the **ODFProcessDist.java** user exit program to customize the ODF output in several ways. You can customize the email attachment or the email notification content. You can customize the print and file output on platforms other than z/OS.

You can customize the ODF output in the following ways:

- ▶ Customize the details and format of the outgoing emails that contain distributions when the value of the distribution location field Location is set to “E-mail” or for any Location value with the “Notify by e-mail” check box selected. For each distribution location type, you can customize the email content and the maximum size for email attachments within a single email.
- ▶ Customize the details of distribution output for all other distribution types on Content Manager OnDemand for Multiplatforms servers, and for all distribution types on z/OS except when the Location value is set to Print. Specify your Simple Mail Transfer Protocol (SMTP) server name to use for outgoing email.
- ▶ Specify whether to enable the Secure Sockets Layer (SSL) when you use the SMTP server to send email.
- ▶ Specify trace parameters.
- ▶ On Content Manager OnDemand for Multiplatforms servers, specify the name of the command to use to submit ODF print requests and the name of the printer queue to use.

The ODFProcessDist.java program uses the ARSODF.XML file as input. The ARSODF.XML file allows customization of the ODF output without modifying the ODFProcessDist.java program. ODF includes a complied version of the sample ODFProcessDist.java program. You can use the sample program as-is, or you can modify the sample program and recompile it to further customize outgoing distribution details.

14.5 Status and monitor tool

The OnDemand Monitor is an interactive workstation client program to check the status of distributions that were submitted for processing and to monitor ODF activity on Content Manager OnDemand servers, beginning at version 9.5. Use this tool to issue a reprint or initiate distributions, as needed.

Figure 14-11 shows the view of the OnDemand Monitor.

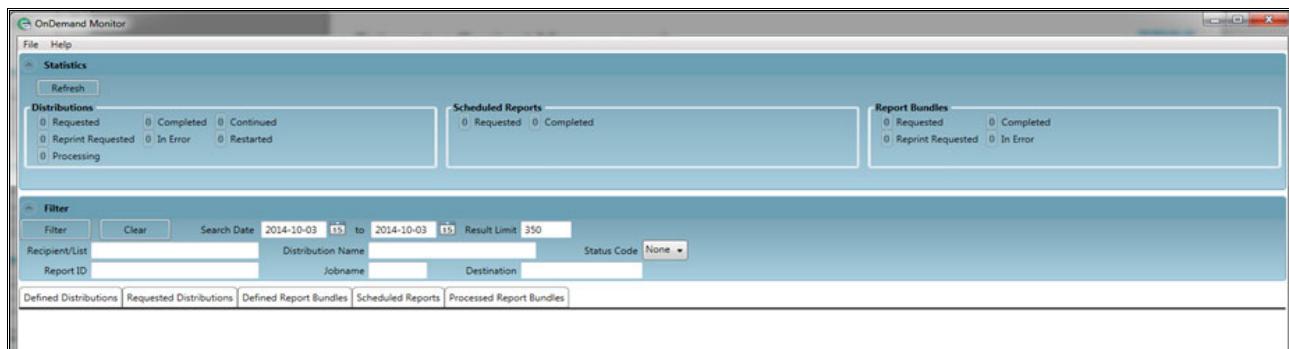


Figure 14-11 Overall view of the OnDemand Monitor

Figure 14-12 on page 332 shows a snapshot of ODF activity.

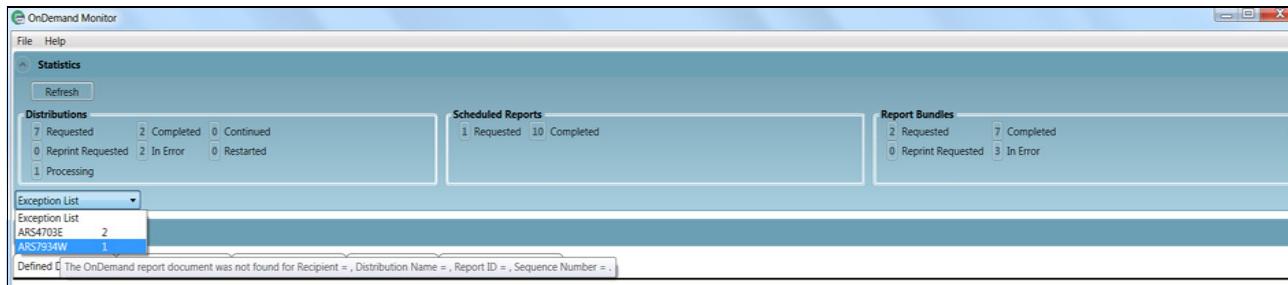


Figure 14-12 Snapshot of ODF activity

The filter reflects the values that were selected to populate the rows in the tabbed section (Figure 14-13).

OnDemand Monitor																	
Statistics																	
Filter																	
Expand or collapse the filter values																	
Search Date 2014-10-14 [15] to 2014-10-14 [15] Result Limit 350																	
Recipient/List																	
Distribution Name																	
Status Code Requested ▾																	
Report ID																	
Jobname																	
Destination																	
Defined Distributions																	
Requested Distributions																	
Defined Report Bundles																	
Scheduled Reports																	
Processed Report Bundles																	
DVD48745	DVD48745 - ALLREADY	2014-10-14 10:25:57.495295	Requested		SCHEDULE	0	Active		All Ready	Print	No	None	Manifest in Sysout	Yes	Wait		
MTSK0002	MTSK0002 - ALLREADY	2014-10-14 11:28:36.961465	Requested		SCHEDULE	0	Active		All Ready	Print	No	None	Manifest in Sysout	Yes	Wait		
MTSK0003	MTSK0003 - ALLREADY W//JOBNAME	2014-10-14 11:31:36.719781	Requested		SCHEDULE	0	Active	MTSK3	All Ready	Print	No	None	Manifest in Sysout	Yes	Wait		
MTSK0012	MTSK0012 - TIME OF DAY	2014-10-14 11:25:883937	Requested		SCHEDULE	0	Active		Time of Day	07:37 Print	No	None	Manifest	Yes	Continu		
MTSK0018	MTSK018A - MAIN TOS	2014-10-14 11:12:57.882988	Requested		SCHEDULE	0	Active	MTSK0018	Time of Day	07:00 Print	No	None	Manifest	No	Wait		
MTSK0036	MTSK0036 - VERIFY ARSRPREA	2014-10-14 11:12:57.881232	Requested		SCHEDULE	0	Active		All Ready	Print	No	None	Manifest in Sysout	Yes	Wait		
MTSK0037	MTSK0037 - VERIFY ARSRPREA W//JOBNAME	2014-10-14 11:12:57.876221	Requested		SCHEDULE	0	Active	MTSK0037	All Ready	Print	No	None	Manifest in Sysout	Yes	Wait		

Figure 14-13 Applying a filter

Figure 14-14 show all of the defined distributions and the most recently requested information.

OnDemand Monitor																	
Statistics																	
Filter																	
Filter																	
Clear																	
Search Date 2014-10-14 [15] to 2014-10-14 [15] Result Limit 350																	
Recipient/List																	
Distribution Name																	
Status Code None ▾																	
Report ID																	
Jobname																	
Destination																	
Defined Distributions																	
Requested Distributions																	
Defined Report Bundles																	
Scheduled Reports																	
Processed Report Bundles																	
BDST0001	BDST0001 RECFM OF DATASETS	2014-10-14 10:25:57.495295	Requested			0	Active		External	File	No	None	No Manifest	No			
BDST0002	BDST0002 - BATCH DIST W//QUERIES					0	Active		External	Print	No	None	No Manifest	No			
BDST0003	BDST0003 - EMAIL					0	Active		Email	Print	Yes	None	Manifest	No			
DVD48745	DVD48745 - ALLSCARW	2014-10-14 10:25:57.495295	Requested		SCHEDULE	0	Active		All Ready	Print	No	None	Manifest in Sysout	Yes			
EXTER	View Distribution Details					0	Active		External	Email	Yes	None	No Manifest	Yes			
MTSK	List Defined Report Bundles					0	Active		Loaded	File	No	None	No Manifest	No			
MTSK	View Distribution Statistics					0	Active		Loaded	Print	No	None	Manifest	Yes			
MTSK	Initiate Distribution					0	Active		SCHEDULE	Print	No	None	Manifest in Sysout	Yes			
MTSK	Cancel Distribution					0	Active		All Ready	Print	No	None	Manifest in Sysout	Yes			
MTSK	Reprint Distribution					0	Active		Loaded	Print	No	None	Manifest	Yes			
MTSK	Edit Output Definitions					0	Active		MTSK0005	Loaded	Print	No	None	Manifest	Yes		
MTSK		2014-10-03 09:13:35.096829	Complete		LOAD	0	Active		MTSK0005	Loaded	Print	No	None	Manifest in Sysout	No		
MTSK0007	MTSK0007 - TIME OF PRINT W//JOBNAME					0	Active		Time of Print	07:45 Print	No	None	Manifest in Sysout	No			
MTSK0007						0	Active		MTSK0007	Time of Print	07:53 Print	No	None	Manifest in Sysout	No		

Figure 14-14 Defined distributions

Figure 14-15 shows all of the distributions that match the filter criteria.

OnDemand Monitor																		
File Help		Statistics		Filter		Defined Distributions Requested Distributions Defined Report Bundles Scheduled Reports Processed Report Bundles												
Recipient/List	Distribution Name	Process Date	Request Status	Reprint	Originator	Continuation Attempts	Defined Status	Jobname	Method	Time	Location	Notify by e-mail	Transform	Manifest	Report Break	Continue		
DVD48745	DVD48745 - ALLREADY	2014-10-14 10:25:57.495295	Requested		SCHEDULE 0	Active	All Ready		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	View Distribution Details	2014-10-14 11:12:57.884885	Error		SCHEDULE 0	Active	All Ready		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	List Included Report Bundles	2014-10-14 11:14:27.887363	Complete		SCHEDULE 0	Active	All Ready		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	View Distribution Statistics	2014-10-14 11:26:22.795675	Complete		SCHEDULE 0	Active	All Ready		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	Initiate Distribution	2014-10-14 11:27.885105	Processing		SCHEDULE 0	Active	MTSK3	All Ready	Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	Cancel Distribution	2014-10-14 11:31:36.719781	Requested		SCHEDULE 0	Active	MTSK3	All Ready	Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK	Reprint Distribution	2014-10-14 11:31:36.719781 [Requested]			SCHEDULE 0	Active	Time of Day 07:37	Print	No	None	Manifest	Yes	Continue					
MTSK	Edit Output Definitions		Reprint fields before reprint		SCHEDULE 0	Active	MTSK0018	Time of Day 07:00	Print	No	None	Manifest	No	Wait				
MTSK	Reprint without any changes				LOAD	Active	Loaded		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK0030	MTSK0030 - VERIFY ARSRPREA W//JOBNAME	2014-10-14 11:25:57.876221	Requested		SCHEDULE 0	Active	All Ready		Print	No	None	Manifest in Sysout	Yes	Wait				
MTSK0037	MTSK0037 - VERIFY ARSRPREA W//JOBNAME	2014-10-14 11:25:57.876221	Requested		SCHEDULE 0	Active	MTSK0037	All Ready	Print	No	None	Manifest in Sysout	Yes	Wait				

Figure 14-15 Requested distributions

Figure 14-16 shows that all of the reports were loaded and that they are available for processing.

OnDemand Monitor																		
File Help		Statistics		Filter		Defined Distributions Requested Distributions Defined Report Bundles Scheduled Reports Processed Report Bundles												
Recipient/List	Distribution Name	Report ID	Process Date	Status	Application Group	Application	Load ID	Document Size	Resource Size	Create Date	C							
EXTE	DAILY SALES EXTERNAL	2014-10-14 11:03:34.452404	Requested	Daily sales external	Daily sales external	78313-1-0-1FAA-0-0-78395	39486	0	2014-10-14 11:03:45.2404	2								
EXTE	View Scheduled Report Details	DAILY SALES EXTERNAL	2014-10-14 14:26:09.657470	Requested	Daily sales external	Daily sales external	78313-1-0-1FAA-0-0-78395	39486	0	2014-10-14 14:26:09.657470	2							
MTSK	List Included Report Bundles	MTSK002A	2014-10-14 11:12:57.884885	Complete	MTSK002A	MTSK002A	78274-1-0-1FAA-20141014000000-20141014000000-78356	42295	0	2014-10-14 11:31:32.904947	2							
MTSK	Display Distribution	MTSK002A	2014-10-14 11:14:27.887363	Complete	MTSK002A	MTSK002A	78274-1-0-2FAA-20141014000000-20141014000000-78356	42295	0	2014-10-14 11:25:07.250590	2							
MTSI	Schedule Distribution	MTSK002A	2014-10-14 11:26:22.795675	Complete	MTSK002A	MTSK002A	78274-1-0-3FAA-20141014000000-20141014000000-78356	42295	0	2014-10-14 11:28:11.199109	2							
MTSI		MTSK002B	2014-10-14 11:26:22.795675	Complete	MTSK002B	MTSK002B	78275-1-0-1FAA-2014050801000000-20050801000000-78357	1163750	0	2014-10-14 11:27:02.495900	2							
MTSK0002	MTSK0002 - ALLREADY	MTSK002C	2014-10-14 11:12:57.884885	Complete	MTSK002C	MTSK002C	78276-1-0-1FAA-20141014000000-20141014000000-78358	1629116	1520415	2014-10-14 11:35:48.41069	2							
MTSK0002	MTSK0002 - ALLREADY	MTSK002C	2014-10-14 11:27.887363	Complete	MTSK002C	MTSK002C	78276-1-0-2FAA-20141014000000-20141014000000-78358	1629116	1520415	2014-10-14 11:52:52.828248	2							
MTSK0002	MTSK0002 - ALLREADY	MTSK002C	2014-10-14 11:26:22.795675	Complete	MTSK002C	MTSK002C	78276-1-0-3FAA-20141014000000-20141014000000-78358	1629116	1520415	2014-10-14 11:28:16.709302	2							
MTSK0003	MTSK0003 - ALLREADY W//JOBNAME	MTSK003A	2014-10-14 11:12:57.885105	Complete	MTSK003A	MTSK003A	78277-1-0-1FAA-20141014000000-20141014000000-78359	42295	0	2014-10-14 11:30:30.81382	2							
MTSK0003	MTSK0003 - ALLREADY W//JOBNAME	MTSK003C	2014-10-14 11:12:57.885105	Complete	MTSK003C	MTSK003C	78279-1-0-1FAA-20141014000000-20141014000000-78361	1629116	1520415	2014-10-14 11:31:16.571036	2							
MTSK0019	MTSK0019 - NO QUERY FOUND	MTSK0019	2014-10-14 13:23:01.474118	Complete	MTSK0019	MTSK0019	78312-1-0-1FAA-20050801000000-20050801000000-78394	1163750	0	2014-10-14 13:30:01.474118	2							

Figure 14-16 Scheduled reports

Figure 14-17 shows all of the report bundles that are being processed or were processed.

OnDemand Monitor																		
File Help		Statistics		Filter		Defined Distributions Requested Distributions Defined Report Bundles Scheduled Reports Processed Report Bundles												
Recipient	Distribution Name	Sequence	Report ID	Process Date	Request Status	Message Num	Jobname	Location	Notify by e-mail	Transform	Application Group	Agid	Application	Aid	Load ID			
MTSK0002	MTSK0002 - ALLREADY	10	MTSK002A	2014-10-14 11:12:57.884885	Error	ARS4703E		Print	No	None	MTSK002A	78274	MTSK002A	78356	78274-1-0-1FAA-20141014000000-201410140000			
MTSK	View Report Bundle Details		MTSK002C	2014-10-14 11:12:57.884885	Error	ARS4703E		Email	Yes	None	MTSK002C	78276	MTSK002C	78358	78276-1-0-1FAA-20141014000000-201410140000			
MTSK	List Included Report Bundles		MTSK002C	2014-10-14 11:12:57.887363	Complete			Print	No	None	MTSK002A	78274	MTSK002A	78356	78274-1-0-2FAA-20141014000000-201410140000			
MTSK	Reprint Report Bundle		MTSK002A	2014-10-14 11:26:22.795675	Complete			Print	No	None	MTSK002A	78274	MTSK002A	78356	78274-1-0-3FAA-20141014000000-201410140000			
MTSK	Reprint Distribution			Edit Output fields before reprint				File	No	None	MTSK002B	78275	MTSK002B	78357	78275-1-0-1FAA-20050801000000-200508010000			
MTSK	Edit Report Bundle Output Definitions			Reprint without any changes				File	No	None	MTSK002B	78275	MTSK002B	78357	78275-1-0-1FAA-20050801000000-200508010000			
MTSK	Display Distribution		MTSK002C	2014-10-14 11:26:22.795675	Complete			Print	Yes	None	MTSK002B	78276	MTSK002C	78358	78276-1-0-2FAA-20141014000000-201410140000			
MTSK	View Message Text		MTSK003A	2014-10-14 11:12:57.885105	Processing		MTSK3A	Print	No	None	MTSK003A	78277	MTSK003A	78359	78277-1-0-1FAA-20141014000000-201410140000			
MTSK0003	MTSK0003 - ALLREADY W//JOBNAME	10	MTSK003C	2014-10-14 11:12:57.885105	Processing		MTSK3C	Email	Yes	None	MTSK003C	78279	MTSK003C	78361	78279-1-0-1FAA-20141014000000-201410140000			
MTSK0019	MTSK0019 - NO QUERY FOUND	10	MTSK0019	2014-10-14 13:23:01.474118	Error	ARS7934W		File	No	None	MTSK0019	78312	MTSK0019	78394	78312-1-0-1FAA-20050801000000-200508010000			

Figure 14-17 Processed report bundles



Full text search

In this chapter, we describe the Full Text Search (FTS) feature of IBM Content Manager OnDemand (Content Manager OnDemand). This feature enables users to build an index of the document content and to search within this full text.

In this chapter, we cover the following topics:

- ▶ Introduction to full text search in Content Manager OnDemand
- ▶ Full text search architecture in Content Manager OnDemand
- ▶ Planning and installing FTS
- ▶ Configuring and operating full text search
- ▶ Running the full text indexing process
- ▶ Using full text search in Content Manager OnDemand clients
- ▶ Troubleshooting tips

15.1 Introduction to full text search in Content Manager OnDemand

Content Manager OnDemand users primarily search on the metadata (extracted index values) that is associated with documents. By using FTS, you can intelligently search through actual document content. To enable FTS, the documents are first parsed and an index is built. This index can then be queried by a full text engine.

The FTS feature in Content Manager OnDemand comes with a new server, the Full Text Search Server (FTS Server), which handles the text extraction, indexing, and searching of the indexed data. This new server offloads the processing of full text data to a machine other than your Content Manager OnDemand library and object servers.

The full text engine is the same search services engine that is used by other IBM products, such as DB2 or IBM FileNet P8. It is based on the Lucene engine and allows advanced and flexible queries. Users can perform wildcard searches, fuzzy (or similar) searches, proximity searches, Boolean searches, and other complex queries.

The full text feature can handle many formats, including Microsoft Office documents, XML files, and typical Content Manager OnDemand formats, such as AFP, Line Data, and Adobe Portable Document File (PDF).

The FTS feature supports full text indexing of both new and existing data. For new data, the FTS Server is configured to index the newly loaded reports by using the Administrator Client. For existing data, indexing is invoked by using the Content Manager OnDemand command-line utilities or the Content Manager OnDemand Web Enablement Kit (ODWEK) Java application programming interface (API).

FTS is enabled through the Content Manager OnDemand folder and allows all clients to take advantage of full text queries after the server configuration is complete. Several new Content Manager OnDemand folder field types are defined in support of FTS. Search score, highlight, and summary are returned, aiding the user in determining whether the document is a good match.

Note: Before the release of the FTS option in Content Manager OnDemand, a document content-based search was possible by using the server-based text search functionality. However, this functionality is limited to AFP, Line, SCS, and PDF documents. It does not use an index, but instead the server retrieves the documents and then scans those documents for the index values. This method limits the capabilities of the functions to exact matches of a query string and might cause workload problems on the Content Manager OnDemand server. FTS eliminates these issues and limitations by introducing new processing components.

15.2 Full text search architecture in Content Manager OnDemand

The process of full text indexing can be lengthy in terms of time and processor consumption. Therefore, an integration architecture, which decouples the full text engine from the Content Manager OnDemand server and keeps the different workloads separate, is required.

The components and their basic communication are shown in Figure 15-1 on page 337.

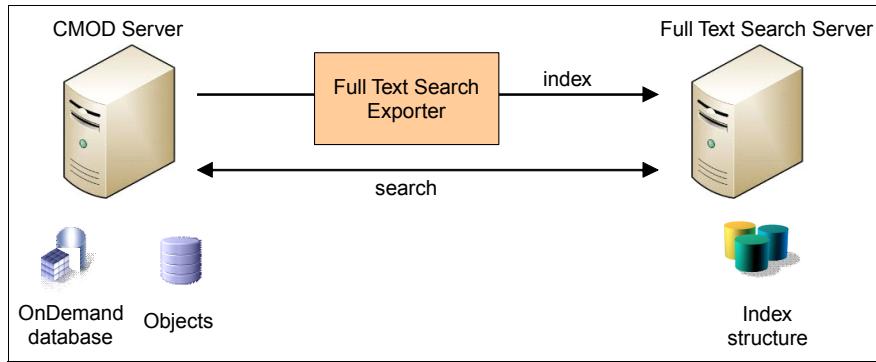


Figure 15-1 Full text search components and communication

15.2.1 Full Text Search Server

The Full Text Search Server (FTS Server) provides a full document processing pipeline that includes text extraction from binary formats, a wide range of encoding support, and language processing in various languages. The flow of data during indexing depends on the configuration and environment. For example, the following process occurs in a single-server configuration:

- ▶ Document contents and properties are sent from the Content Manager OnDemand repository through FTS Exporter to the FTS Server.
- ▶ FTS then preprocesses the data, including text extraction, language identification, tokenization, and language analysis on the documents.
- ▶ After preprocessing, the document content is stored in the FTS index.

The FTS Server comes with text extractors for many varied document types, including Microsoft Office formats and XML. However, for AFP and Line Data, text extraction occurs within the FTS Exporter. Images do not contain text, and they are not supported in FTS.

15.2.2 Index structure

The FTS Server creates a binary Lucene index that is stored on the FTS Server. The index is segmented into logical groupings called *collections*. The segmentation model is designed to parallel the data table segmentation model in Content Manager OnDemand. For each application group data table, which has data that is indexed in the FTS Server, a collection is created on the FTS Server. Therefore, FTS collections maintain a one-to-one relationship with Content Manager OnDemand data tables. Collections are created with the following naming convention:

InstanceName_TableName

This naming convention allows the FTS index to scale horizontally without affecting the performance of the Content Manager OnDemand server. During a query operation, you can narrow the scope of documents that must be searched. If the user specifies a date range in addition to the FTS criteria, the Content Manager OnDemand segment tables are referenced to determine the collections that must be queried.

Because the full text index contains the processed text of each indexed document, the index can become large. For more information about size calculation, see 15.3.6, “Index considerations” on page 341.

15.2.3 Indexing document through FTS Exporter

New documents that are to be full text indexed are retrieved from Content Manager OnDemand by the FTS Exporter component. These documents are then pushed into the full text engine of the FTS Server. The detailed process is shown in Figure 15-2.

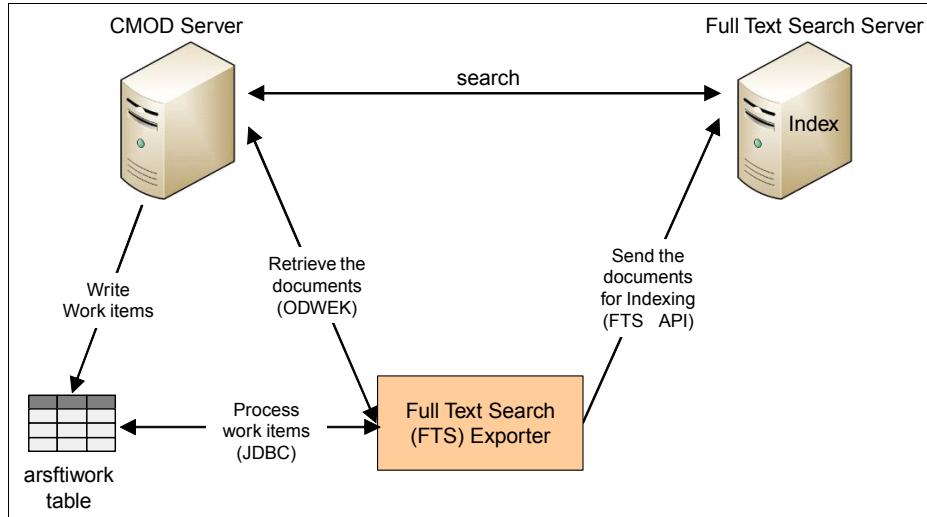


Figure 15-2 The role of the FTS Exporter in full text indexing

To support the FTS Server, a new table (*arsftiwork*) was created in the Content Manager OnDemand database. The *arsftiwork* table is used to hold full text indexing work items. Whenever a new document will be indexed, a work item record is created in the *arsftiwork* table as part of the Content Manager OnDemand load process. For existing data, this process occurs explicitly by using the command-line tools or the ODWEK Java API.

To index new documents, the FTS Exporter connects to the *arsftiwork* table and works through the records, retrieving the associated documents from the Content Manager OnDemand server and pushing them into the FTS Server to be indexed. Documents that are to be removed from the full text index follow the same process. The FTS Exporter handles all tasks that relate to adding, updating, and deleting documents to and from the full text index.

The FTS Exporter application is a Java application that communicates with the Content Manager OnDemand server to retrieve the documents from the Content Manager OnDemand server and push the documents to the FTS.

15.2.4 Searching

Search queries are handled by the Content Manager OnDemand server by directly communicating with the FTS Server. When an FTS string is specified during a query in a Content Manager OnDemand folder, a query is issued to the FTS Server for all applicable collections that match the date range. If no date range is specified in the query, all collections for the specified application group are queried.

15.3 Planning and installing FTS

The following section describes the main aspects of the FTS component installation and configuration.

15.3.1 Component overview

FTS in Content Manager OnDemand consists of the FTS Server, the Full Text Search Exporter (FTS Exporter), and a Content Manager OnDemand server that uses both components to provide FTS to the users.

Full Text Search Server

The FTS feature in Content Manager OnDemand is a separately licensed component that must be downloaded and installed. It contains the FTS Server. Full text Indexing and Search functionality can be implemented on any Content Manager OnDemand platform (z/OS, IBM i, and Multiplatform). The FTS Server itself runs only on Multiplatforms systems. The FTS Server is typically installed on a different system than the Content Manager OnDemand server because of the difference in workload types and the amount of processing that is required for high performance and throughput.

Full Text Search Exporter

The FTS Exporter is a Java application, which is available as a JAR file (`ODFTIExporter.jar`), that comes with the Content Manager OnDemand server installation (starting with version 9.0). The `ODFTIExporter.jar` file is in the `jars` subdirectory.

The FTS Exporter relies on the following components:

- ▶ Java Database Connectivity (JDBC) database drivers for your Content Manager OnDemand database (DB2, Oracle, or SQL Server on Windows).
- ▶ Java Runtime Environment (JRE) (Java 1.7.0) or later can be used to run the `ODFTIExporter.jar` file.

The FTS Exporter communicates with the Content Manager OnDemand server to retrieve the documents that are sent to the FTS Server. It uses a JDBC connection to the Content Manager OnDemand database to read the `arsftiwork` table.

The FTS Exporter can be run on the Content Manager OnDemand server system or from any other system that is connected by TCP/IP. The FTS Exporter does not require the existence of the Content Manager OnDemand database on the same system. The FTS Exporter obtains the instance configuration from the Content Manager OnDemand server.

For more information, see 15.4.2, “Configuration of the Full Text Search Exporter” on page 344.

Note: Ensure that you apply the latest Content Manager OnDemand version and fix pack to the Content Manager OnDemand server and the FTS Server component before you use FTS.

15.3.2 Installing the FTS Server

Install the FTS Server on a Multiplatforms system by running the FTS Server setup program. Use the command-line parameter `-i console` for a console mode setup.

The setup creates a set of directories under the `FTS_Home` (installation target) directory. Most of these directories are not modified after the installation.

Special attention is required for the following directories:

- ▶ bin: Contains all of the executable files
- ▶ config: Contains the configuration and the index structures
- ▶ log: The log files of the FTS Server

Ensure that the target location has sufficient free disk space for the log files (at least 100 MB). Otherwise, the FTS Server stops logging and returns an error code. For more information about capacity planning for the config directory and the index size, see 15.3.6, “Index considerations” on page 341.

15.3.3 Operating system resources

For better throughput results during the indexing process on AIX, Linux, and Solaris servers, ensure that the operating system resource limits are set correctly.

The values of the **fsize** (maximum file size) and **nfiles** (maximum number of files that are allowed for a process) parameters must be set to unlimited (-1) or 65536 to ensure correct system operation. The FTS Server startup script checks these settings and tries to correct them for the session. They can be set permanently by modifying the /etc/security/limits or /etc/security/limits.conf files.

15.3.4 Workload

Processor consumption depends on the following items:

- ▶ Number of collections
- ▶ Number of documents for each collection
- ▶ Number of concurrently indexed collections
- ▶ Required indexing throughput
- ▶ Query load

For more information, see the capacity planning topics in the introduction and planning guides for Multiplatforms and z/OS in the IBM Knowledge Center:

- ▶ http://www.ibm.com/support/knowledgecenter/SSQHWE_9.5.0/com.ibm.ondemandtoc.doc/planning.htm
- ▶ http://www.ibm.com/support/knowledgecenter/SSEPCD_9.5.0/com.ibm.ondemandtoc.doc/planning.htm

A minimum of one processor, 2 GB of RAM, and 8 GB of swap space must be assigned to the FTS Server. For more information, see *Hardware and Software Requirements*:

<http://www.ibm.com/support/docview.wss?rs=129&uid=swg27021456>

15.3.5 Memory heap size

During indexing and searching, the FTS Server consumes heap memory for storing the indexed documents, preprocessing and indexing queues, and indexing memory structures. To optimize the performance of the FTS Server, it is important that the maximum heap memory size in the Java virtual machine (JVM), the queue size, and file size limits are configured correctly. You can configure the maximum heap size by using the configuration tool.

The **maxHeapSize** parameter sets the maximum heap size for the FTS Server. The default is 1.5 GB. This value must be a number between 1.5 GB and the amount of available memory.

The maximum file size that can be processed is a function of the heap size. When you set the maximum heap size to a value greater than 2 GB, the file size limits for text, XML, and binary documents must be increased for new collections. For each 8.3 MB of heap memory over 2 GB, the values of the file size limits (60 MB by default) must be increased by 1 MB (up to 400 MB), as demonstrated by the following formula:

$$\text{Max file size} = 60 \text{ MB} + (\text{heap memory} - 2 \text{ GB}) / 8.3$$

For example, a 2 GB maximum heap size results in 60 MB as the maximum size of a file that can be processed.

15.3.6 Index considerations

The most significant sizing option for the FTS Server system is the hard disk requirements for the full text index. The FTS Server requires a fast disk subsystem. Because the textual representation of each indexed document is stored in the disk subsystem, a considerable amount of disk space might be needed.

Index size calculation

Although the disk space usage depends on the text in each document, this usage is linear to the original size of the indexed data. Typically, the size of the index on the disk is 50% - 150% of the original text size as illustrated in the following formulas:

$$\text{minimum disk space} = \text{Number of documents} \times \text{document size} \times 50\%$$

$$\text{maximum disk space required} = \text{Number of documents} \times \text{document size} \times 150\%$$

The actual percentage, 50% through 150%, is data-dependent. So, an exact number can be obtained only by testing with your data.

For example, 100,000 documents of 20 KB each can require about 1500 MB (100,000 x 20 KB x 75%) of disk space.

Tip: To determine the text size for AFP and Line Data documents, extract a sample document and use the `arsview` server command to determine the text size.

The size of the index is not limited. However, when data is added to or removed from a text index, the text index structure is merged to improve query performance. The required processing time to complete the merger depends on several factors, such as index size and absolute throughput (which in turn depends on the data type and index format). These factors result in practical limits on the total text index size.

For query performance, the biggest impact is the number of matching results, not the size of the text index.

Temporary disk storage

During the indexing process, the server requires additional disk space for temporary storage. The maximum required disk space is approximately four times the total size of the text of the documents that are indexed.

Index location

The full text index is stored within the installation directories of the FTS Server. The default directory is shown:

`<FTS_Home>/config/collections/<collection_name>/data/text`

If you want to place the configuration and the index structures into a different file system path, use the **configTool** command-line utility in the FTS_Home/bin directory. You must perform this action immediately after the installation, that is, before you start the FTS Server and create any full text indexes by using Content Manager OnDemand. After an index is created, the index structures cannot be changed.

The configuration and index location are stored in the **defaultDataDirectory** parameter. First, show the current value of the parameter by running the following command:

```
configTool.sh list -system -defaultDataDirectory
```

Then, you can change the value by running the following command:

```
configTool.sh set -system -defaultDataDirectory <new value>
```

On Windows platforms, **configTool.sh** is available as **configTool.cmd**.

After you change the **defaultDataDirectory** parameter, you must restart the FTS Server.

15.4 Configuring and operating full text search

The FTS Server can be operated by the startup and shutdown scripts in the FTS_Home/bin subdirectory. The FTS Server must be running to perform indexing and full text searches.

After the FTS Server is started, by default it listens on TCP port 8191. Content Manager OnDemand and the FTS Exporter must know this port to communicate with the FTS Server. The port can be changed by using the port parameter with the **configTool**. For more information about how to use this command, see “Index location” on page 341.

The following command-line tools are installed in the bin directory and used to manage the FTS Server:

- ▶ **adminTool**: Used to manage collections, set trace options, and check statuses
- ▶ **configTool**: Used to review and change most system and collection parameters
- ▶ **startup** and **shutdown** scripts
- ▶ **stopwordTool**: Used to add or modify the list of stop words (common words that are not indexed)
- ▶ **synonymTool**: Used to add or remove synonym dictionaries from the index
- ▶ **dumpIndex**: Used to dump documents from the index

15.4.1 Base configuration in Content Manager OnDemand

To enable FTS in Content Manager OnDemand, FTS must be enabled for each of your Content Manager OnDemand instances. In Windows, you enable FTS for each of your Content Manager OnDemand instances in the Content Manager OnDemand Configurator by selecting the **Enable Full Text Index and Search** check box on the Server (Advanced Options) window.

On all other platforms, the ars.cfg file of your Content Manager OnDemand instance must be edited. You must add the following line:

```
ARS_SUPPORT_FULL_TEXT_INDEX=1
```

You must restart the instance after this configuration to enable the FTS option in the Content Manager OnDemand Administrator. Then, you can configure the FTS options on the application groups and folders.

Configuring application groups for full text search

FTS support must be configured for each application group for which you plan to perform full text index and search.

To FTS-index an application group, configure the application group for FTS by completing the following steps:

1. Click **Application Group** → **General tab** → **Advanced** and select **Yes** under Full Text Index. Specify the FTS Server name and port. The default port is 8191. Choose whether to automatically index all new loads. Figure 15-3 shows these settings. The setting **Full Text Index documents automatically** indexes new documents after they are loaded.

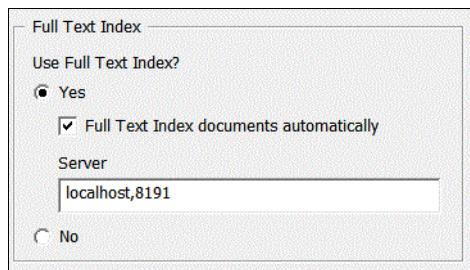


Figure 15-3 Enable full text indexing in an application group

2. Add an FTS field to the application group in the Field Definition tab. (Its name does not matter.) On the Field Information tab, set the field data type to **Small Int (2)** and select the **Full Text Index attribute** option.
3. Modify the permissions and add the **Full Text Index** permission to users and groups who must be able to index documents (users who load and run a full text indexing request through `arsdoc` or the API).

Configuring a folder for full text search

The Content Manager OnDemand folders must be configured before any full text searching can occur. Four new folder field types were added to support FTS:

- ▶ **Full Text Index Search**

This field is required for users to specify their FTS criteria. This field can be queried only.

- ▶ **Full Text Index Score**

This field is optional. It represents the score of the hit, relative to the other matching hits. It can be queried and displayed in the hit list.

- ▶ **Full Text Index Highlight**

This field is optional. It returns the text that surrounds the matching text. It represents the context in which the text was found. This field can be only displayed in the hit list. Highlighting is not supported for XML documents.

- ▶ **Full Text Index Summary**

This field is optional. It returns the first 80 characters of the document. This information might be useful, depending on the data. For example, bills and statements typically have identical text for headers; therefore, this information cannot be used to distinguish hits.

15.4.2 Configuration of the Full Text Search Exporter

FTS Exporter requires configuration parameters for connecting to Content Manager OnDemand, its database, and the FTS Server. These parameters can either be specified on the command line or written into a configuration file. We recommend that you use the config file because your JDBC connection password is part of the required parameters and stored with the other password parameters that are encrypted in the config file.

To create the config file, run the FTS Exporter with the **configure** parameter:

```
Java -jar ODFTIExporter.jar configure -configFile <file>  
<all configuration parameters>
```

The following parameters are required:

-dbEngine <db engine>	DB engine (DB2, MSSQL, or ORACLE)
-dbHostname <server>	Database server host name
-dbPort <port>	Database port
-dbUser <user>	Database user ID
-dbPassword <passwd>	Database password
-dbName <db name>	Database name
-dbOwner <db owner>	Database owner
-odInstance <instance>	
-odUser <user>	OnDemand user ID
-odPassword <password>	OnDemand user password
-odInstallDir <path>	Where OnDemand is installed
-pollDelay <seconds>	Number of seconds between polling (optional)
-ftiToken <FTI authentication token>	Optional
-tempDir <path>	Temporary work directory (optional)
-traceDir <path>	Directory to store trace files (optional)
-traceLevel <export trace level>	

Table 15-1 describes the purpose of each parameter.

Table 15-1 FTS Exporter parameters

Parameter	Purpose
dbEngine	The engine of the database that is being used. This parameter defines the JDBC class that is used by the FTS Exporter. It must be either DB2, MSSQL, or ORACLE.
dbHostname	The host name of the database server that runs the Content Manager OnDemand instance database.
dbPort	The port number of the database server.
dbUser, dbPassword	The user and password combination that is used to connect to the database.
dbName	For Multiplatform systems, this value is the database name of the instance database to connect to. For DB2 on z/OS, this value is the database location. For IBM i, the dbName is the instance name.

Parameter	Purpose
dbOwner	The database owner (used to open the correct arsftiwork table).
odInstance	The Content Manager OnDemand instance to connect to. This parameter must match the Content Manager OnDemand instance name that is in the ars.ini file (or registry in Windows).
odUser, odPassword	The user and password of a Content Manager OnDemand user. This user is used to retrieve the documents for full text indexing.
odInstallDir	The installation directory of the Content Manager OnDemand server. This server contains the ars.cfg file, which is used to look up the instance name.
pollDelay	Optional. A polling interval in seconds in which the FTS Exporter checks the arsftiwork table for new work items.
ftiToken	Optional. The Full Text Index (FTI) authentication token that is used to communicate with the FTS Server.
tempDir	Temporary directory.
traceDir, traceLevel	The location of trace files and the tracing level. If the tracing level is specified, it must be any of the following tokens: OFF, SEVERE, WARNING, INFO, FINE, FINER, or FINEST.

A call to configure the FTS Exporter is similar to the call that is shown in Example 15-1.

Example 15-1 Configuring the FTS Exporter

```
java -jar ODFTIExporter.jar configure -configFile odfts.cfg -dbEngine DB2
-dbHostname localhost -dbPort 60004 -dbUser ondemand -dbPassword ondemand -dbName
ondemand -odInstallDir /opt/ibm/ondemand/V9.0 -pollDelay 60 -tempDir /tmp
-traceDir /tmp -ftiToken "fIqBxTQ=" -odUser admin -odPassword ondemand -dbOwner
ondemand -odInstance ONDEMAND
```

Example 15-1 writes the configuration file odfts.cfg and configures a connection to the Content Manager OnDemand instance ONDEMAND with the user admin and to the Content Manager OnDemand instance DB2 database ondemand. The FTS Exporter polls for work items in the arsftiwork table every 60 seconds and processes them against the FTS Server that is configured with this Content Manager OnDemand instance.

Content Manager OnDemand supports running the FTS Exporter on a system other than your library and object server. In certain instances, this separation is highly recommend. If the FTS Exporter is installed on a remote system, the Content Manager OnDemand server code must be installed on this system because the FTS Exporter requires part of the binary and supporting files from the Content Manager OnDemand server installation. The FTS Exporter also gets part of its connection information from the ars.ini file that is installed with the Content Manager OnDemand server or from the Windows registry.

The JDBC connection user that is used by the FTS Exporter must have SELECT, UPDATE, and DELETE authority on the arsftiwork table and SELECT authority on the arsseg table of each Content Manager OnDemand instance it works with.

To connect to your Content Manager OnDemand database by using JDBC, additional driver JAR files are required. The FTS Exporter is built to reference two additional JAR files in its directory by default: jdbc1.jar and jdbc2.jar. To use JAR file execution capability, you must link (or copy) your required JDBC driver JAR files to these locations so that they are automatically loaded by the FTS Exporter.

For example, when you use DB2 on AIX or Linux, you can issue the following commands in the jars subdirectory of the server to create these two links:

```
ln -s /opt/IBM/db2/V9.7/java/db2jcc.jar  jdbc1.jar
```

```
ln -s /opt/IBM/db2/V9.7/java/db2jcc_license_cu.jar  jdbc2.jar
```

For the connection with Content Manager OnDemand, the FTS Exporter automatically references the Java API 0DApi.jar.

Note: Each instance of the FTS Exporter can connect to one Content Manager OnDemand instance. Only a single instance of the FTS Exporter for each Content Manager OnDemand database is supported.

15.5 Running the full text indexing process

Both the FTS Server and the FTS Exporter must be running for the full text indexing requests to complete.

If the FTS Exporter and the FTS Server are not running, all full text indexing requests are written to the arstiwork table. They are not processed until the FTS Exporter processes them and sends the documents to a running FTS Server.

15.5.1 Automatically indexing new data during the load

Indexing new data is simple with Content Manager OnDemand and FTS. When FTS is configured correctly, the result of an **arsload** operation automatically creates work items in the arstiwork table. Each application group that the FTS is enabled for must be configured correctly. For more information, see “Configuring application groups for full text search” on page 343.

15.5.2 Indexing existing data through the **arsdoc** command

The **arsdoc** command is enhanced with two new options. The first new option is **fti_add**. Parameters for this option control whether the resulting documents are queried through SQL (the **-i** parameter) or if an entire load will be full text indexed (the **-X** parameter).

The second new option that is added to **arsdoc** is **fti_release**. This option takes the same parameters as **fti_add** to determine the documents that need their indexes removed from the full text index.

Both of these options result in work items that are created in the arstiwork table. Example 15-2 shows an example of the command that is used with Bank1.

Example 15-2 Full text indexing all (SQL WHERE 1=1) documents of Bank1

```
arsdoc fti_add -f "Bank1" -h localhost -i "where 1=1" -u admin -v -G Bank1
```

15.5.3 Indexing existing data through ODWEK

The ODWEK Java API contains two new methods that support FTS. The first method is `ODFolder.FTIAddHits()`. This method has a single parameter, which is a Vector of ODHits. The Vector of ODHit objects can be produced by using the `search()` methods of the ODFolder. All hits that are contained in the Vector parameter to `FTIAddHits()` are sent to FTS Exporter for full text indexing.

The second new method of the ODFolder object is `FTIReleaseHits()`. This method also takes a Vector of ODHit objects as a parameter. This method is used to remove the indexes from the FTS Server.

Both of these calls produce work items in the arsftiwork table.

For more information about the ODWEK Java API, see *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646.

15.5.4 Running the FTS Exporter

The FTS Exporter processes the work items in the arsftiwork table. The FTS Exporter begins processing work items, starting with the oldest items. It continues to process these work items until the table is empty. Then, it goes to sleep for a specified amount of time before it wakes up to look for more work items.

After you configure the FTS Exporter, as described in 15.4.2, “Configuration of the Full Text Search Exporter” on page 344, you must run the FTS Exporter with the config file as a parameter. The FTS Exporter requires a reference to the ODWEK native libraries that ship with Content Manager OnDemand to work correctly. The easiest way to perform this task is to add this reference to the start command line when you run Java with the FTS Exporter JAR file.

Example 15-3 shows how to start the FTS Exporter with `odfts.cfg` as the configuration file and `/opt/ibm/ondemand/V9.0/lib64` as the directory where the native library is installed.

Example 15-3 Running the FTS Exporter

```
java -Djava.library.path=/opt/ibm/ondemand/V9.0/lib64 -jar ODFTIExporter.jar index  
-configFile odfts.cfg
```

In Windows environments, ensure that you enclose the ODWEK path with quotation marks if it contains spaces.

15.6 Using full text search in Content Manager OnDemand clients

All Content Manager OnDemand clients use the same process and procedure when they search the full text index. The query is first sent to the Content Manager OnDemand server for processing. If the application group that is being searched contains a segment date, and if the search criteria specified a date range, that range is used to narrow the collections on the FTS Server that must be searched.

15.6.1 Syntax

The FTS Server supports a rich query language that enables fuzzy searches, proximity searches, weighted searches, and Boolean searches.

Queries can contain terms and operators. A *term* is a single word, such as “united”. A *phrase* is a group of words that are contained in quotation marks, such as “computer software”. Phrases are searched as exact expressions. Stop words are not removed. No lemmatization is performed. Boolean operators, such as AND, or wildcards, such as asterisk (*) or question mark (?), are treated as literal characters.

Without quotation marks, the query is parsed and the syntactical options that are described in the following sections are allowed.

15.6.2 Boolean searches

Boolean operators allow terms to be combined through logical operators. The following Boolean operators are supported: AND, OR, NOT, and the minus sign (-).

Boolean operators must be specified in all uppercase characters. For example, when you search for documents about dogs or cats while you specify the OR Boolean operator, specify the query as dogs OR cats, not dogs or cats.

Precede a term with a minus sign (-) to indicate that the term must be absent from a document for a match to occur. For example, the following query returns documents that include the term computer and not the term hardware:

```
computer -hardware
```

Use parentheses to control the Boolean logic in a query. For example, the following query finds documents that contain either WebSphere or IBM Lotus and website:

```
(WebSphere OR Lotus) AND website
```

15.6.3 Wildcard searches and optional terms

FTS supports wildcard searches. You can place wildcard characters before, within, or after a term.

Use a question mark (?) to perform a single character wildcard search. For example, the following query finds documents that contain the terms mare, mere, mire, and more:

```
m?re
```

Use an asterisk (*) to perform a multiple character wildcard search. A multiple character wildcard search looks for zero or more alphanumeric characters. The following query finds documents that contain the terms bar, rebar, far, afar, and car:

```
*ar
```

Use a percent sign (%) to indicate that a search term is optional. For example, the following query finds documents that include the term log and optionally include the term file:

```
log %file
```

15.6.4 Fuzzy and proximity searches

A fuzzy search query searches for character sequences that are not only the same but similar to the query term. Use the tilde symbol (~) at the end of a term to perform a fuzzy search. For example, the following query finds documents that include the terms analytics, analyze, and analysis:

```
analytics~
```

An optional parameter can be used to specify the required similarity. Specify a value greater than 0 and less than 1. The value must be preceded by a 0 and decimal point, for example, 0.8:

```
analytics~0.8
```

A value closer to 1 matches terms with a higher similarity. If the parameter is not specified, the default is 0.5.

A proximity search finds documents that contain terms within a specified number of words of each other. Use the tilde symbol (~) to perform a proximity search. For example, the following query finds documents that contain “IBM” and “WebSphere” within seven words of each other:

```
"IBM WebSphere"~7
```

Proximity search is supported for individual terms, not phrases. Also, a word after a sentence break is considered 10 positions apart from the last word of the previous sentence.

15.6.5 Weighted searches (boosting terms)

Follow a search term by a boost value to influence how documents that contain a specified term are ranked in the search results. Use the caret symbol (^) with a number (the boost factor) at the end of the term. For example, the following query finds documents that include the terms IBM and Germany and increases the relevance of these documents by a factor of five in the search results:

```
ibm Germany^5.0
```

Note: Special characters, such as punctuation marks, are not alphanumeric characters. They are not supported in fuzzy or proximity searches, or they are not hit by a wildcard (*) operator.

15.7 Troubleshooting tips

If you encounter any problems during full text indexing and searching, investigate the issue by looking at the different logs or by using the trace options.

15.7.1 Content Manager OnDemand server log

Each full text indexing operation is registered at the Content Manager OnDemand system log. Example 15-4 on page 350 lists a few message numbers with their text.

Example 15-4 Content Manager OnDemand system log messages that relate to full text search

Message 397: Document Full Text Index Add: App1GroupName(Adobe PDFs) Agid(5021)
Full Text Index Notified(1) Count(16) Time(0.069)

Message 398: Document Full Text Index Add Failed: App1GroupName(Adobe PDFs)
Agid(5021) Full Text Index Notified(0) Count(16) Time(0.001)

Message 399: Document Full Text Index Delete: App1GroupName(Adobe PDFs) Agid(5021)
Full Text Index Notified(1) Count(16) Time(0.025)

Message 226: Application Group Query: Name(BaxterBayBank) Agid(5025) Time(0.120)
Hits(2) Count() Sql(WHERE ODDAT_Sdate BETWEEN '1996-06-22' AND '2013-06-22')
SqlR() FullTextSearch(lunch* newark) FullTextScore() ServerTextSearch() AnnColor()
AnnText() OrderBy()

Message 439: FTS Error: IQQS0032E The query lunch~x cannot be processed because it
has incorrect syntax. Causes of the problem: IQQP9014E The query [lunch~x]
cannot be parsed because there is a syntax error at position 7. The fuzzy argument
value [x] is not valid because its data type is not float or double. --
File=arsfti.cpp, Line=394

Messages 397, 398, and 399 are viewable and contain the list of documents (their metadata) that are affected by this operation. In the case of message 398 (fail), the failure reason is documented, as well.

Each time the FTS Server reports an error, message 439 is issued, and it contains the error message that was returned by the FTS Server. In this case, the query that is entered by the user contained a wrong syntax for a proximity or fuzzy search.

15.7.2 Full Text Search Server log

You can troubleshoot the FTS Server by configuring and viewing the FTS Server logs. The FTS Server generates logging information during server startup, indexing, and searching. The log files contain configuration information, warnings, errors, and debugging information that can be useful for monitoring the server and for troubleshooting specific issues. The command-line tools also generate log files. By default, log files are stored in the FTS_Home/log directory. You can run the **configTool** with the **list -logFolder** command to see your log directory.

Every message in the log file has an associated level that indicates the message type. The logging levels, in descending order of severity, are defined:

- ▶ SEVERE: Errors and exceptions that occur while the server is running. Typically, SEVERE messages include detailed information with the stack trace.
- ▶ WARNING: Mild problems that might require the attention of an administrator, such as a missing value for a setting with a default value, or the truncation of a document during indexing.
- ▶ INFO: Informational messages that are generated during system operation.
- ▶ FINE: Detailed messages for debugging purposes. This level includes parsed queries.
- ▶ FINER: More details, for example, the results of document parsing.
- ▶ FINEST: The most detailed level.

The default logging level is INFO, which means that messages of levels SEVERE, WARNING, and INFO are generated. To view the current logging level, run the FTS Server **adminTool** with the **printLogLevel** command. To change the logging level at run time, run the **adminTool** with the **configureTrace -logLevel** command.

15.7.3 Full Text Search Exporter trace

When you experience issues with the FTS Exporter, enable tracing by using the **-traceDir** and **-traceLevel** parameters. Set the **-traceLevel** parameter to FINE when you troubleshoot a problem. A trace file is created and named `ftiexport_0.0_DDMMYYHHMM.log` in the directory that is specified by the **-traceDir** parameter.

By enabling trace within the FTS Exporter, you also enable trace in ODWEK, which results in the creation of an ODWEK trace file that is named `arswww.trace`. This trace file is also written to the `traceDir` directory. The FTS Exporter trace files can be read with any text editor, but the ODWEK trace files are viewable only by using the Content Manager OnDemand **arstfmt** command.

If you are running the FTS Exporter by using a configuration file, you must create a separate configuration file with the trace level set to a different level because the command-line parameter is ignored when you use a configuration file.

15.7.4 Authentication and FTS Exporter errors

Use the following tips to help you troubleshoot authentication and FTS Exporter-related errors.

Authentication errors

If you encounter any errors about authentication in the FTS Exporter trace, the FTS Server log, or message 439 in the Content Manager OnDemand system log with the following message text, the wrong authentication token might be in use:

```
FTS Error: IQQD0040E The client specified the wrong authentication token. --
File=arsfti.cpp, Line=394
```

The default authentication token of `fIqBxTQ=` can change because of a reinstallation of the FTS Server or other severe incidents. You can discover the current authentication token by running the **configTool** of the FTS Server with the **printToken** parameter. See Example 15-5.

Example 15-5 Displaying the active token

```
# /opt/ibm/odfts/V9.0/bin/configTool.sh printToken
The authentication token is printed below. This token is used to communicate with
the server. Store the token if applicable.
fIqBxTQ=
```

You can configure the authentication token that is used by Content Manager OnDemand through a configuration setting in the `ars.cfg` file of your instance:

```
ARS_FULL_TEXT_INDEX_TOKEN=fIqBxTQ=
```

You can also configure the configuration parameter and the parameter file of the FTS Exporter application through a configuration setting in the `ars.cfg` file of your instance.

If you are document the default token (for example, as a parameter of the FTS Exporter), be aware that the second character of the token value is an uppercase letter I, as in IBM.

FTS Exporter errors

If you encounter issues with the FTS Exporter, increase the trace level as described in 15.7.3, “Full Text Search Exporter trace” on page 351. Also, ensure that you review the configuration file by opening it in a text editor. Check whether all settings are correct.

If you encounter errors about ars3wapi in the FTS Exporter output or trace, the FTS Exporter cannot find the native library reference to the ODWEK system libraries. To see how to use the **-D** parameter of Java to include the native library path at application start, see 15.5.4, “Running the FTS Exporter” on page 347.

If the error relates to a `java.lang.UnsatisfiedLinkError` error that does not find the `ars3wapi32`, you are running on a 32-bit JVM. The Java classes of the FTS Exporter try to load the `libars3wapi64`, which is in the `lib64` subdirectory of your Content Manager OnDemand installation. If they cannot load the `libars3wapi64`, a 32-bit version is searched (which is not present in the `lib64` folder). If both attempts fail, they fail with the related error message.

Important: Ensure that you run the FTS Exporter with a 64-bit JVM that can load the 64-bit share library `libars3wapi64`.

For more information about ODWEK native libraries, see “Accessing the native libraries” on page 195.



Enhanced Retention Management

In this chapter, we describe Enhanced Retention Management, which is a feature of IBM Content Manager OnDemand.

In this chapter, we cover the following topics:

- ▶ Enhanced Retention Management overview
- ▶ Configuring Enhanced Retention Management
- ▶ Applying and releasing holds
- ▶ Enhanced Retention Management use cases

16.1 Enhanced Retention Management overview

The Content Manager OnDemand Enhanced Retention Management feature helps you manage and enforce the retention of documents in a Content Manager OnDemand system. In a Content Manager OnDemand system, you retain documents for a specific period. Retaining documents for a certain period is commonly referred to as *retention management*. *Records management* describes the process of retaining and deleting documents under a set of circumstances that are not necessarily bound by time, for example, until the end of litigation. Consider the following important points when you select the correct retention solution:

- ▶ Records management requires that you can control individual documents. Content Manager OnDemand manages application groups, not individual documents, and it works with a storage manager to delete (expire) documents.
- ▶ Records management requires flexibility in defining when to delete documents. Content Manager OnDemand defines the time to delete applications with fixed time ranges, for example, five years after Content Manager OnDemand loads the documents.

To overcome these limitations, implement the Content Manager OnDemand Enhanced Retention Management feature. With the Enhanced Retention Management feature, you control individual documents by introducing *holds*. Holds are a way to identify the documents that you want to keep for a period of time. To expire the document, you must remove the hold. Holds give you the flexibility to choose when to delete documents because you control when to remove a hold. You can manage holds through any of the following interfaces:

- ▶ Content Manager OnDemand Windows client, Content Navigator, WEBi, the **ARSDOC** command, or the Content Manager OnDemand Web Enablement Kit (ODWEK) Java APIs that are provided by Content Manager OnDemand.
- ▶ IBM FileNet P8, when integrated with Content Manager OnDemand by enabling the FileNet Content Federation Services for Content Manager OnDemand feature. Use Content Federation Services for Content Manager OnDemand to federate Content Manager OnDemand repositories, which connect your Content Manager OnDemand content to business processes in the IBM Case Foundation and IBM Enterprise Records features of FileNet P8.

The Enhanced Retention Management feature requires that you disable the expiration processes that the Content Manager OnDemand storage manager has on the documents that you want to hold. Disabling the expiration process prevents the Content Manager OnDemand storage manager from deleting these documents.

16.2 Configuring Enhanced Retention Management

You must configure Content Manager OnDemand to identify documents in application groups that you want to retain (hold) and also identify the users who can manage holds. You must disable expiration processes by the storage manager so that it cannot expire data. You must also convert application groups with an expiration type of DOCUMENT, SEGMENT, or STORAGE MANAGER to an expiration type of LOAD.

To configure Enhanced Retention Management, you must perform the following tasks:

- ▶ Enable Enhanced Retention Management
- ▶ Identify the application groups where Enhanced Retention Management is applied
- ▶ Specify the application group “lockdown” field
- ▶ Enable hold permission for the application group
- ▶ Assign hold permissions to users

- ▶ Create holds by using the Administrator Client
- ▶ Configure a folder to display that a document is held

These items are described in more detail in the following sections.

16.2.1 Enabling Enhanced Retention Management

To enable Enhanced Retention Management, modify the ars.cfg file and add the following line:

```
ARS_SUPPORT_HOLD=1
```

In Content Manager OnDemand for Windows, you can enable Enhanced Retention Management by using the Content Manager OnDemand configurator, as shown in Figure 16-1. Select the **Enable Enhanced Retention Management** check box.

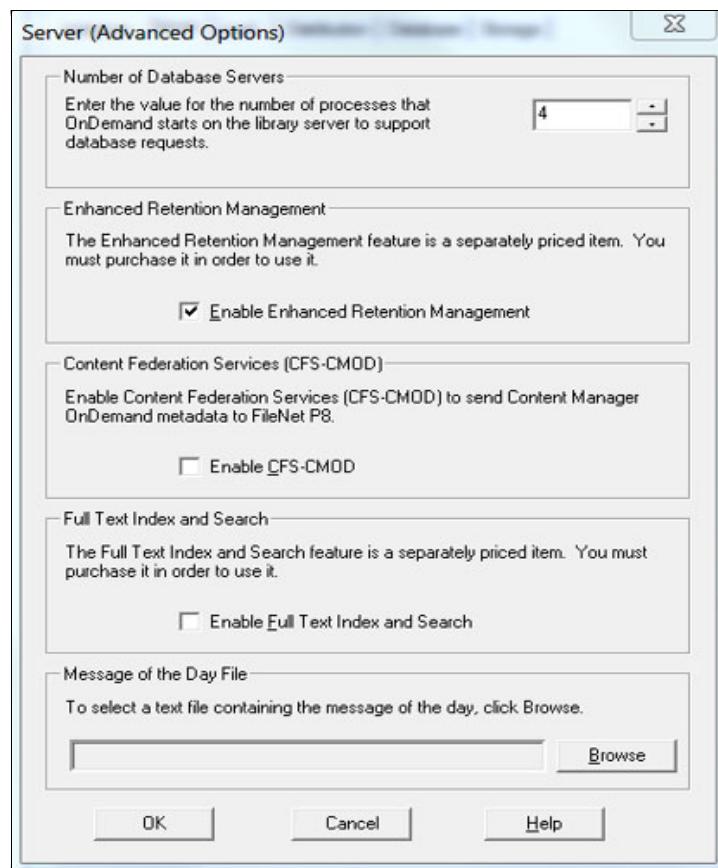


Figure 16-1 Enabling Content Manager OnDemand for retention management

Clearing this configuration setting has no effect on any documents that were placed on hold by Enhanced Retention Management. Documents continue to be held until all holds are removed.

16.2.2 Identify the application groups

By using the Content Manager OnDemand Administrator, you specify whether you want to use Enhanced Retention Management at the application group level, as shown in Figure 16-2 on page 356.

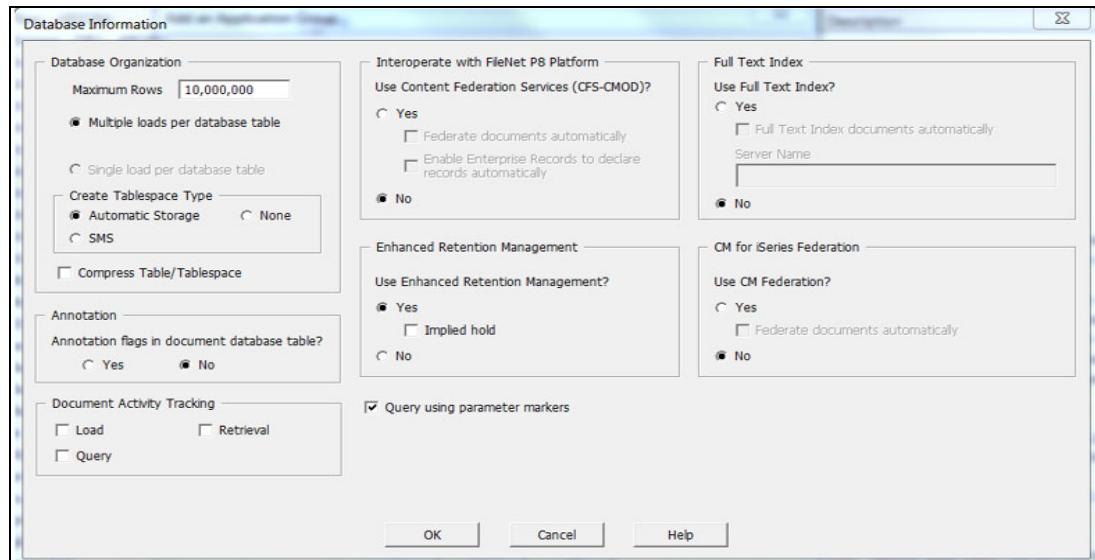


Figure 16-2 Configuring the application groups

When you create an application group, the default for Enhanced Retention Management is No. By selecting Yes, you can also define whether you want to define implied holds. By selecting an implied hold, all of the data that is loaded into the application group is inherently held. The implied hold on the documents within the application group can be removed only by calling either **ARSDOC** or the ODWEK Java API to remove the implied hold. Implied holds are used in solutions where the retention of documents is maintained outside of Content Manager OnDemand.

16.2.3 Specify the application group lockdown field

When you use Enhanced Retention Management, you must define an application group field, as shown in Figure 16-3 on page 357. This field must be marked as **Lockdown**, and it is used to maintain the hold status of individual documents. To mark the field as the Lockdown type, you must select **Small Int (2)** for the Data Type of the field.

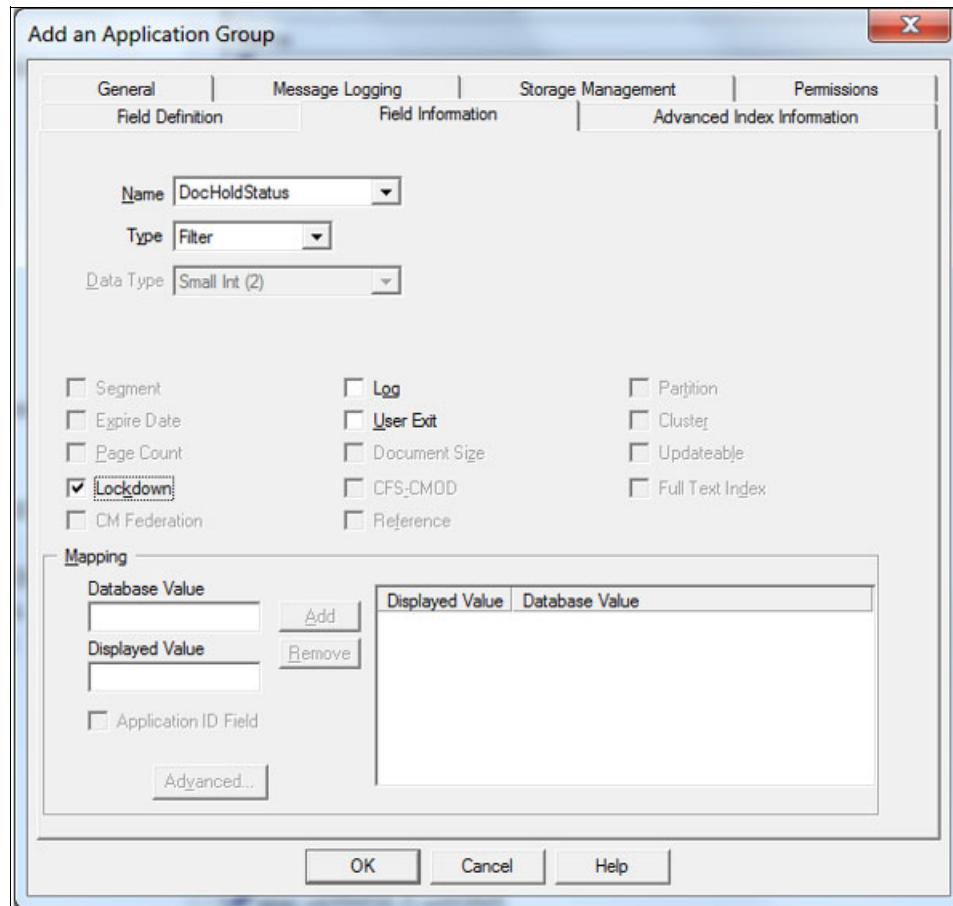


Figure 16-3 Configuring the application group Lockdown field

The Lockdown field is used to maintain a count of the number of holds on the document. If no holds exist, its value is 0. If an implied hold exists for the document, the value is 16384. Any additional holds that are applied or released increase or decrease this number by 1.

16.2.4 Enabling hold permission for the application group

To apply a hold to a document or to release a document, you must have the appropriate permissions. A new permission of type Hold exists under the permissions for a document in the application group permissions, as shown in Figure 16-4 on page 358.

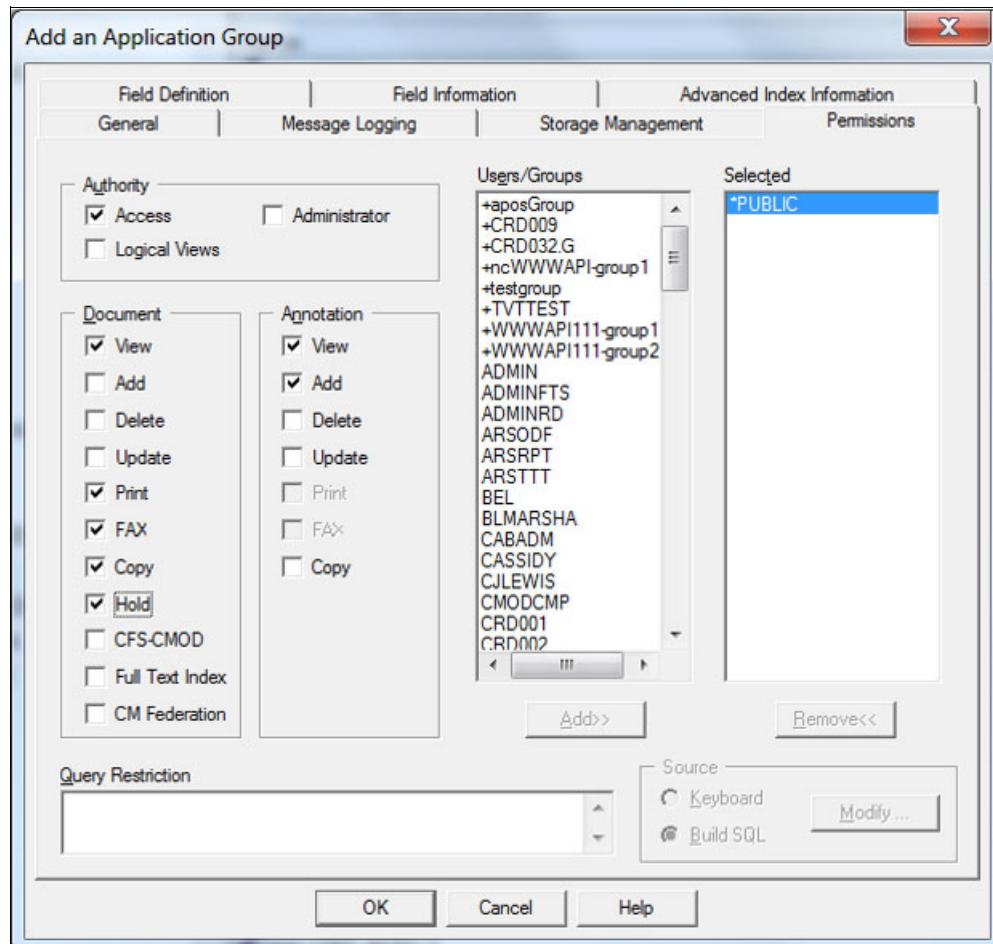


Figure 16-4 Configuring application group permissions

16.2.5 Assigning hold permissions to users

To manage holds in Content Manager OnDemand, you must have the appropriate permission. Figure 16-5 on page 359 shows you how to define a user type of Hold Administrator and to provide an authority type of Create Holds.

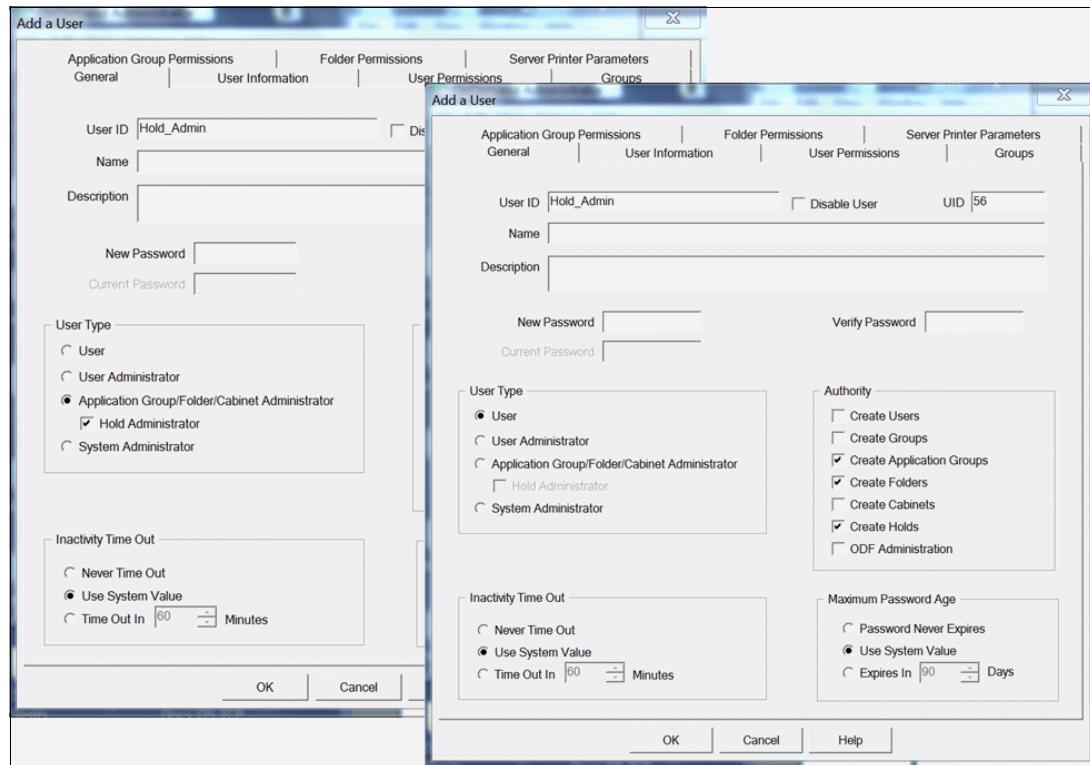


Figure 16-5 Define a user with hold entitlements

16.2.6 Creating holds by using the Administrator Client

Hold definitions can be created and removed by using the Administrator Client. To define the hold definition, go to the Administrator Client, select the **Holds** tab, and provide a name and description for the hold that you are creating under the General tab (see Figure 16-6 on page 360). Select the **Permissions** tab to grant the permissions that are required for your hold.

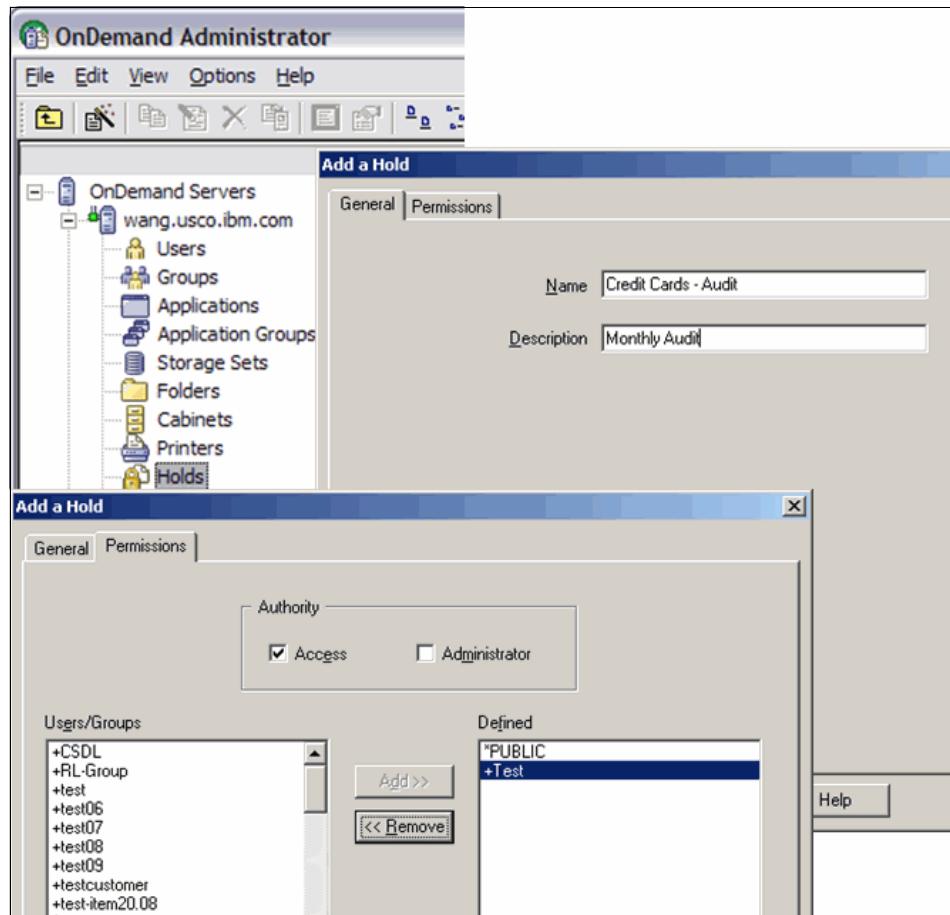


Figure 16-6 Configuring holds

16.2.7 Configuring a folder to display that a document is held

At the folder level, a new check box was added so that you can specify whether you want to see an icon in the client that indicates whether a document is locked down. Figure 16-7 on page 361 shows you how to configure a folder for hold capabilities.

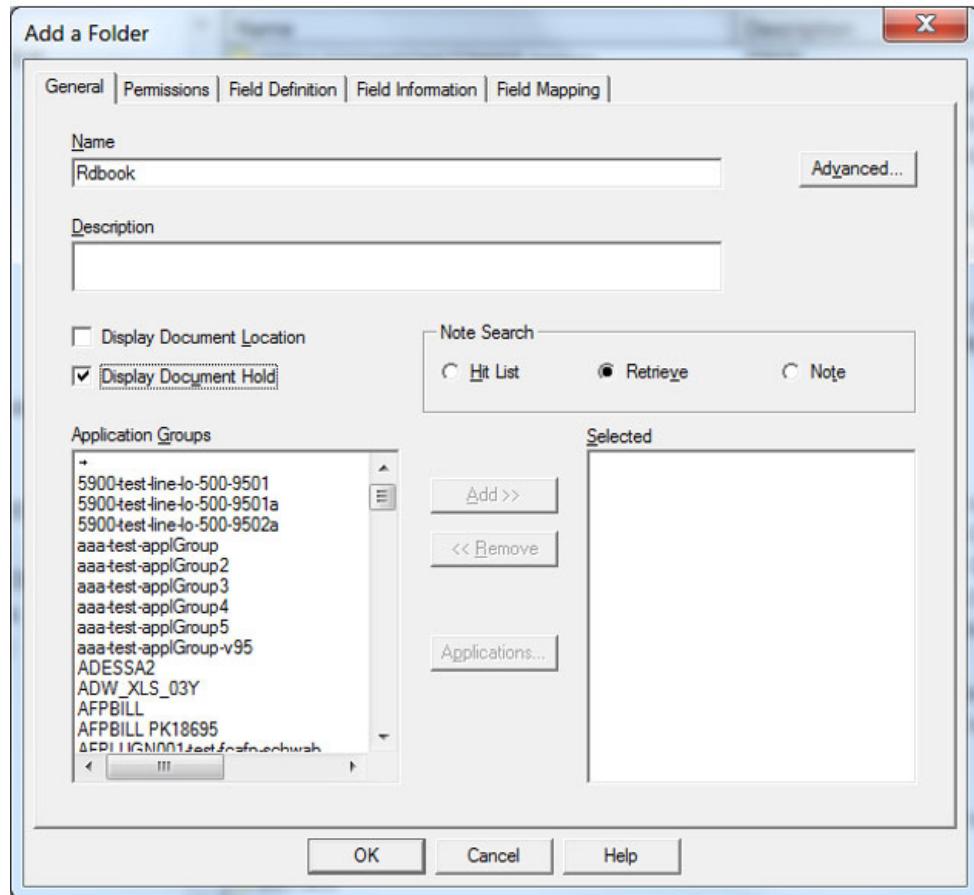


Figure 16-7 Configuring a folder

16.3 Applying and releasing holds

Use the Enhanced Retention Management feature to apply and release holds quickly and efficiently. When you apply a hold to a document or multiple documents, they cannot be expired. If your documents are part of a large batch load, only the documents in a hold status are expired when they reach their expiration date. After a document is released from a hold status, it can be expired based on its original retention management policy.

16.3.1 Managing holds

You can apply and release holds by using the Windows client, Content Navigator, WEBi, the ODWEK Java API, and ARSDOC. As shown in Figure 16-8 on page 362, when you select documents from the hit list and right-click, the options for applying, releasing, and showing holds display.

16.3.2 Applying holds

Users can apply holds to a document or documents that are defined to an application group with the enabled Enhanced Retention Management feature. To apply a hold, select a document and click **Actions** → **Holds** → **Apply Hold**, as shown in Figure 16-8 on page 362.

This action prevents the deletion of documents in Content Manager OnDemand regardless of the documents' expiration dates.

The screenshot shows the IBM Content Navigator interface. On the left, there's a sidebar with 'CMODServer' and a search bar labeled 'Name contains'. Below it is a 'Recent Searches' section with a tree view. The tree has a node 'Student Bills - Student Bills' which is expanded, showing several document items like 'Loan Delinquency Reports - I', 'Scanned Image Data - Scannr', etc. Other collapsed sections include 'All Searches' and 'DemoCabinet'. The main area is titled 'Student Bills' and shows a table of search results. The table has columns for 'ID', 'Student Name', and 'Date'. There are 15 rows of data. A context menu is open over the first row, with the 'Actions' dropdown expanded. The menu includes options like 'Open', 'Preview', 'Download', 'Properties', 'View Notes', 'Holds', 'Print', 'Send Email', and 'Export'. The 'Holds' option is further expanded, showing 'Apply Hold' and 'Remove Hold', both of which are highlighted with a blue selection bar.

ID	Student Name	Date
78	ARCHIVE, PENELOPE	09/15/11
78	ARCHIVE, PENELOPE	09/15/11
98	LEK, CARL	09/15/11
98	LEK, CARL	09/15/11
98	BOHACEK, KAREN	09/15/11
98	BOHACEK, KAREN	09/15/11
225 37 2878	ERICKSON, PETE	09/15/11
225 37 2878	ERICKSON, PETE	09/15/11
242 87 5208	SCHILLING, JEFF	09/15/11
242 87 5208	SCHILLING, JEFF	09/15/11
258 43 3292	GARCIA, THOM	09/15/11

Figure 16-8 Highlighting and selecting documents to apply holds

16.3.3 Creating and removing custom holds

Users can assign previously created holds to documents or create a hold that can be used privately or publicly, as shown in Figure 16-9.

The screenshot shows a modal dialog box titled 'Apply Hold'. Inside, there's a section labeled 'Available holds:' containing a list box. The list box shows three items: 'Demo', 'Demo Holds', and 'Ed's Hold', with 'Ed's Hold' currently selected. At the bottom of the dialog are three buttons: 'New Hold', 'Apply Hold', and 'Cancel'.

Figure 16-9 Creating custom holds

When you want to remove a previously enabled hold, highlight the documents and click **Actions → Holds → Remove Hold**, as shown in Figure 16-10 on page 363.

Note: When a document is on hold, a hold icon is displayed to the left of the document.

The screenshot shows the IBM Content Navigator interface. On the left, there's a sidebar titled 'CMODServer' with sections for 'Recent Searches' (including 'Student Bills - Student Bills', 'Loan Delinquency Reports - I', etc.) and 'All Searches'. The main area is titled 'Student Bills' and shows a search results table with columns for 'Date' and 'ID / Student Name'. One row in the table has a 'Hold' icon to its left. A context menu is open over this row, with the 'Holds' option expanded to show 'Apply Hold' and 'Remove Hold' options. The table data is as follows:

Date	ID / Student Name
09/15/11	78 ARCHIVE, PENELOPE
09/15/11	78 ARCHIVE, PENELOPE
09/15/11	LEK, CARL
09/15/11	LEK, CARL
09/15/11	BOHACEK, KAREN
09/15/11	BOHACEK, KAREN
09/15/11	ERICKSON, PETE
09/15/11	ERICKSON, PETE
09/15/11	SCHILLING, JEFF
09/15/11	SCHILLING, JEFF
09/15/11	GARCIA, THOM

Figure 16-10 Removing holds

16.3.4 Search for hold documents

Hold capabilities were developed as part of the Content Manager OnDemand database structure as a field that can be indexed. Users can use this capability to search on selective holds that might be applied by another user, as shown in Figure 16-11.

The screenshot shows the IBM Content Navigator interface with a search query of 'Loan Delinquency Reports x Student Bills x'. The search criteria panel is expanded, showing filters for 'Student ID', 'Student Name', 'Date', and 'Holds'. The 'Holds' filter dropdown is open, showing three entries: 'Demo', 'Demo Holds', and 'Ed's Hold'.

Figure 16-11 Holds as a searchable index

16.4 Enhanced Retention Management use cases

Enhanced Retention Management provides two methods that allow users to place documents in a hold status:

- ▶ Ad hoc: Use the ad hoc method to put documents in a hold status selectively based on their search criteria.
- ▶ Load: The load method places documents in a hold status at load time.

The following sections present two use cases that provide examples of these methods.

16.4.1 Ad hoc holds

A healthcare firm receives frequent ongoing requests to find documents that must be gathered and protected from deletion because of a legal case that involves the firm's customers. Historically, retention management was a paper-based world of file folders, filing cabinets, and off-site warehouses with elaborate and costly storage requirements. Retention management changed dramatically with the rapid proliferation of digital information.

The challenges of locating and retaining documents quickly was a problem for the firm until they set up Content Manager OnDemand as its standard tool for capturing documents. Content Manager OnDemand can capture high volumes of content and quickly search and retrieve documents with multiple client solutions for both desktop and standard web browsers. By using Content Manager OnDemand, the firm was able to enable the Enhanced Retention Management feature and have a business solution to select, hold, and prevent expiration of documents easily within seconds. The return on investment (ROI) easily justified the investment in the Enhanced Retention Management feature.

16.4.2 Load holds

A financial firm has a business requirement to review transactions for the day before they release the transactions for viewing by their customer base. Content Manager OnDemand is used as the enterprise report capture solution. The firm wants to use Content Manager OnDemand to build a simplistic workflow. The solution also must be able to put the documents back in a hold status quickly to prevent any expiration rules that are assigned by the Content Manager OnDemand application from running.

By enabling the Enhanced Retention Management feature and its “hold on store” capability, the firm has a solution that meets its business requirements. By placing every document on hold in a specific application group in Content Manager OnDemand, the process is controlled by entitlements. User A (the reviewer) has administrator rights to apply and remove holds on documents. After the document is reviewed, User A releases the hold, which signifies to User A’s application area that the document was reviewed.



Content Federation Services for Content Manager OnDemand and IBM Enterprise Records

In this chapter, we describe how to enable records management for an IBM Content Manager OnDemand (Content Manager OnDemand) solution. By default, report and document expiration are controlled by the storage managers that are integrated with Content Manager OnDemand. By using the storage managers, you can assign a retention period at data capture time. IBM Enterprise Records enhances retention capabilities with the flexibility to assign event-based retention and make a report or document an official compliant record to meet numerous government regulations.

In this chapter, we cover the following topics:

- ▶ Content Federation Services for Content Manager OnDemand and IBM Enterprise Records overview
- ▶ Administration of Content Federation Services for Content Manager OnDemand for Enterprise Records
- ▶ Content Federation Services for Content Manager OnDemand architecture
- ▶ Deployment considerations

17.1 Content Federation Services for Content Manager OnDemand and IBM Enterprise Records overview

IBM FileNet Content Federation Services enables organizations to access content from numerous heterogeneous repositories anywhere in the enterprise and federate this information to provide a single enterprise source for critical business content. Content Federation Services for Content Manager OnDemand enables enterprises to perform federation, search, retrieve, and records management functions across Content Manager OnDemand repositories.

IBM Enterprise Records positions your business to provide legally compliant records that meet government regulations at the time of inquiry that follow your corporate record policy file plan.

Content Manager OnDemand handles a high volume of document ingestion to the system, typically of a static nature, such as credit card or bank statements. Each document ingestion might contain thousands of individual documents or pages. Content Manager OnDemand offers a retention feature so that you can set the document retention for a fixed period at the document ingestion time, for example, an investment company that applies a simple retention policy of eight years to all of their customer statements.

Content Manager OnDemand does not apply an event-based retention policy that is based on, for example, the date that the customer closed an account. In this scenario, the clock does not begin the eight-year period until the customer closes the account. By enabling records federation services by using Content Federation Services for Content Manager OnDemand, you can manage Content Manager OnDemand content in a manner that is consistent with your organization's records retention policies.

When Content Manager OnDemand content is federated and declared as a record in Enterprise Records, Content Manager OnDemand content can be tied to dynamic retention policies, such as account closure, policy termination, contract execution, or any other event. In these circumstances, records federation services can allow your organization to retain content for a certain amount of time, starting on the date of the event. Companies must design their policies carefully to manage a large collection of data correctly when the company deals with various regulatory policies and litigation.

When it is time to expire data, with federated and declared content by using Content Federation Services for Content Manager OnDemand, Content Manager OnDemand can delete the original load (which contained multiple documents) and at the same time reingest those documents that must be retained.

When you use Enhanced Retention Management, you enable the holding documents immediately and prevent expiration. Although this feature is powerful, it does not position your business to make a Content Manager OnDemand captured report or document a compliant record to meet government regulations. You must enable the feature that meets your business requirements:

- ▶ Use Enhanced Retention Management in Content Manager OnDemand to hold documents and prevent expiration.
- ▶ Use Enterprise Records to make documents into compliant records and to enable them for event-based expiration.

A situation might exist where both features are enabled because of many different line-of-business requirements.

17.2 Administration of Content Federation Services for Content Manager OnDemand for Enterprise Records

Configure Content Manager OnDemand for Content Federation Services to declare records by using Enterprise Records. You must disable expiration processes by the storage manager so that it cannot expire data. You must also convert application groups with an expiration type of DOCUMENT, SEGMENT, or STORAGE MANAGER to an expiration type of LOAD.

To configure Content Federation Services for Content Manager OnDemand, you must perform the following tasks:

- ▶ Enable Content Federation Services for Content Manager OnDemand.
- ▶ Identify the application groups where Content Federation will be enabled.
- ▶ Specify the application group field.
- ▶ Enable Content Federation permissions for the application group.
- ▶ Federate document metadata to Content Federation Services for Content Manager OnDemand.

These items are discussed in more detail in the following sections.

17.2.1 Enabling Content Federation Services for Content Manager OnDemand

All of the steps in this section assume that IBM FileNet P8 and FileNet Content Federation Services are installed correctly.

In this section, we describe the components in Content Manager OnDemand to enable the federation capabilities to allow record declaration in Enterprise Records. We assume that you are familiar with Content Manager OnDemand administration, so detailed steps are not provided in this chapter.

For more information about the installation and configuration of FileNet P8 and FileNet Content Federation Services, see *Federated Content Management: Accessing Content from Disparate Repositories with IBM Content Federation Services and IBM Content Integrator*, SG24-7742.

To use IBM FileNet P8 Content Federation Services for Content Manager OnDemand, you must enable the feature in Content Manager OnDemand by modifying the ars.cfg file and adding the following line:

```
ARS_SUPPORT_CFSOD=1
```

In Content Manager OnDemand for Windows, you can enable IBM FileNet P8 Content Federation Services for Content Manager OnDemand by using the Content Manager OnDemand Administrator Client Configurator. Figure 17-1 on page 368 shows the Content Manager OnDemand configuration setup for Content Federation Services for Content Manager OnDemand.

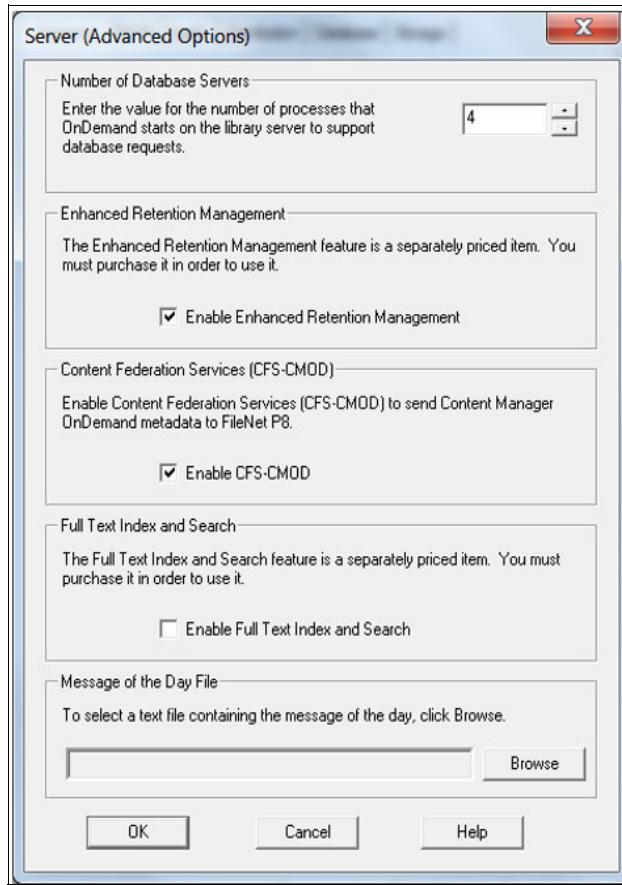


Figure 17-1 Configuration setup for Content Federation Services for Content Manager OnDemand

Disabling this configuration setting does not affect any existing documents that were placed on hold by Enterprise Records. Documents continue to be held until Content Manager OnDemand is notified by Enterprise Records that the documents must be deleted.

17.2.2 Identify the application groups where Content Federation will be enabled

For each application group, specify whether you want to enable FileNet P8 Content Federation Services for Content Manager OnDemand by using the Content Manager OnDemand Administrator Client, as shown in Figure 17-2 on page 369.

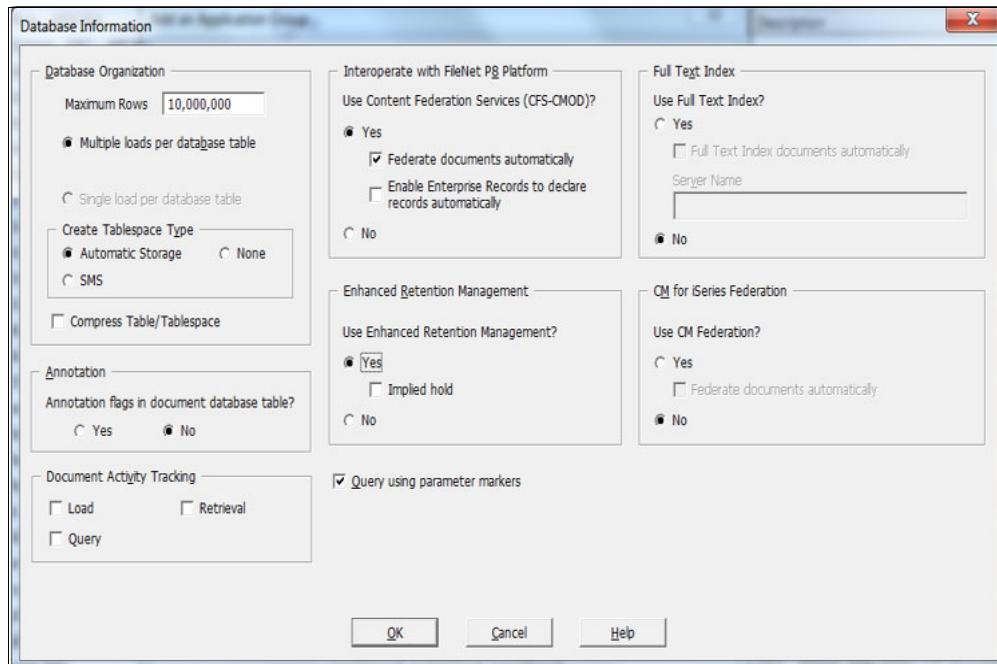


Figure 17-2 Configuring an application group to enable Content Federation Services

When you create an application group, Content Federation Services for Content Manager OnDemand (CFS-CMOD) is disabled, by default. To enable, select **Yes** for the “Use Content Federation Services (CFS-CMOD)” option. Optionally, you can define whether you want to federate documents automatically and whether to enable Enterprise Records (previously known as IBM FileNet Records Manager) automatically.

Selecting “Federate documents automatically” means that when data is loaded into Content Manager OnDemand, the metadata also is sent to Content Federation Services for Content Manager OnDemand to be made available to FileNet P8.

The “Enable Enterprise Records to declare records automatically” option is used as a performance option. Setting this flag means that Content Manager OnDemand assumes that Enterprise Records now controls expiration processing. Not setting this option means that every time a document is put under Enterprise Records control, FileNet P8 then notifies Content Manager OnDemand to lock down the document, which might result in many requests to Content Manager OnDemand. By choosing to perform this task automatically, Content Manager OnDemand can avoid this impact.

17.2.3 Specifying the application group field

When you use FileNet P8 Content Federation Services for Content Manager OnDemand, you must define an application group field, which must be marked as **CFS-CMOD**, as shown in Figure 17-3 on page 370. To mark the field as the CFS-CMOD type, you must select **Small Int (2)** for the Data Type field. This field is used to maintain the Content Federation Services for Content Manager OnDemand status of individual documents. Figure 17-3 on page 370 shows how to configure the application group field to enable Content Federation Services for Content Manager OnDemand.

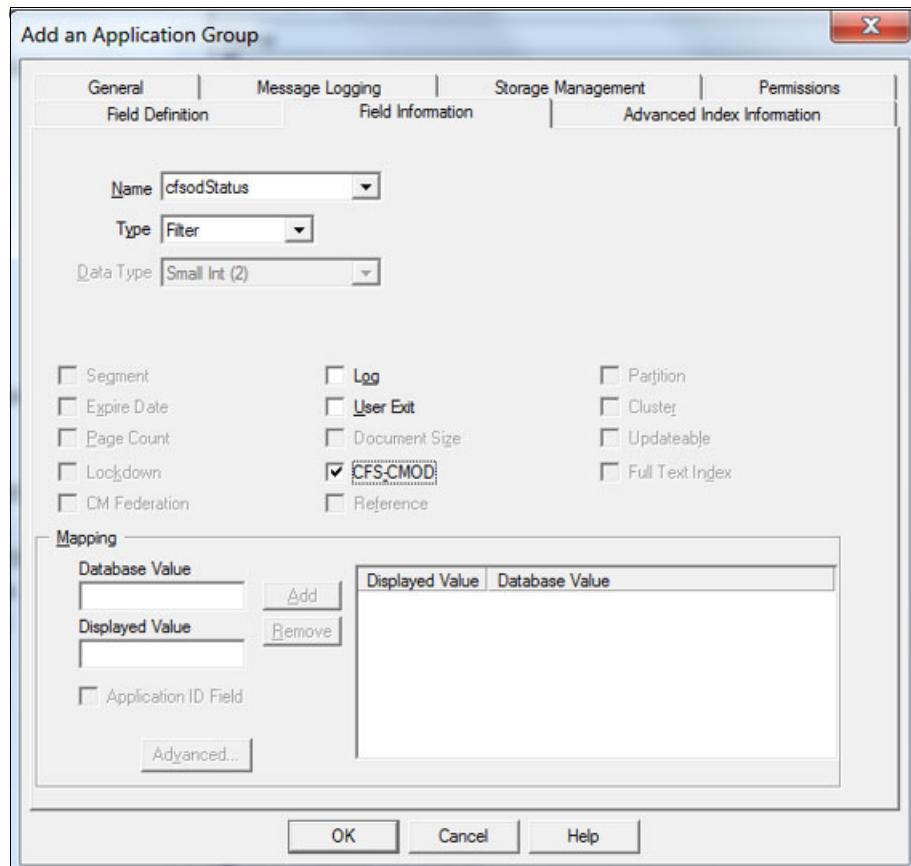


Figure 17-3 Configuring the application group field

The CFS-CMOD field is used to maintain the status of Content Federation Services for Content Manager OnDemand. If the document is not given to Content Federation Services for Content Manager OnDemand, the value is zero. If the value is negative, the document is eligible for deletion. Otherwise, the flag is a logical OR flag with the following values:

- ▶ 0x0001: If federated to Content Federation Services for Content Manager OnDemand
- ▶ 0x1000: If declared in Enterprise Records
- ▶ 0x2000: If the metadata for the document cannot be updated

17.2.4 Enable Content Federation permissions for the application group

You must have the appropriate permission to federate a document. A new permission of type CFS-CMOD now exists under the permissions for documents in the application group permissions, as shown in Figure 17-4 on page 371.

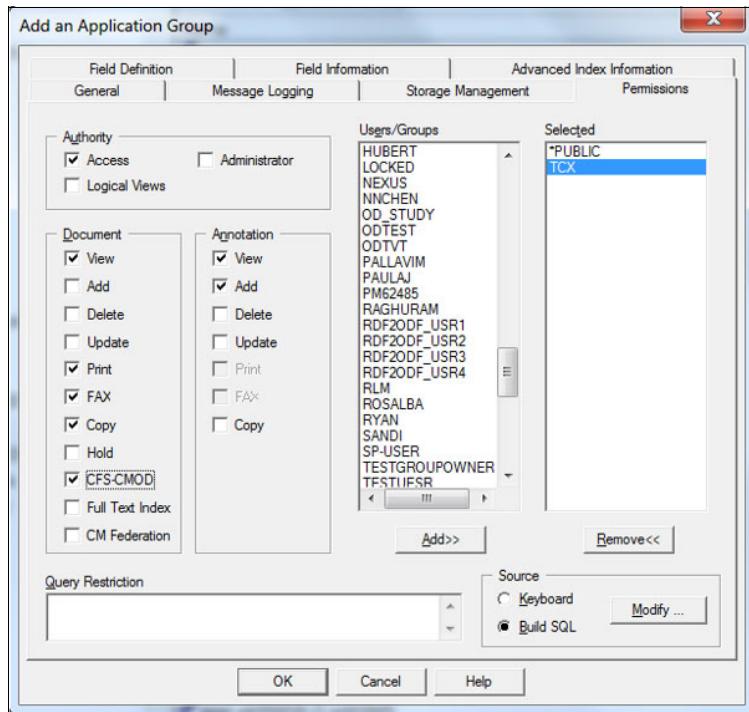


Figure 17-4 Defining permissions to enable CFS-CMOD

17.2.5 Federating document metadata to Content Federation Services for Content Manager OnDemand

Two steps are required to federate Content Manager OnDemand documents to FileNet P8:

1. Mark the documents by using the Content Manager OnDemand Windows Client, ODWEK Java APIs, or ARSDOC. If the option to federate documents automatically is set, the documents are marked automatically when they are loaded.
2. Push the metadata of the documents to FileNet P8 by using the Content Federation Services for Content Manager OnDemand Exporter utility.

Important: After you push the document metadata by using the Content Federation Services for Content Manager OnDemand Exporter utility, you must not change the application group ID or application ID of the Content Manager OnDemand source. They are part of the document identifier of the virtual document that was created in FileNet P8. As the result, the previously federated document's content is no longer accessible from FileNet Content Platform Engine.

For more information about this task, see 4.5.4, “Configure and run the CFSOD exporter utility”, in *Federated Content Management: Accessing Content from Disparate Repositories with IBM Content Federation Services and IBM Content Integrator*, SG24-7742.

If the option to automatically federate is not set, data still can be federated by using the Windows client, ODWEK Java APIs, or ARSDOC. Figure 17-5 shows where you can select an option to federate selected metadata to FileNet P8.

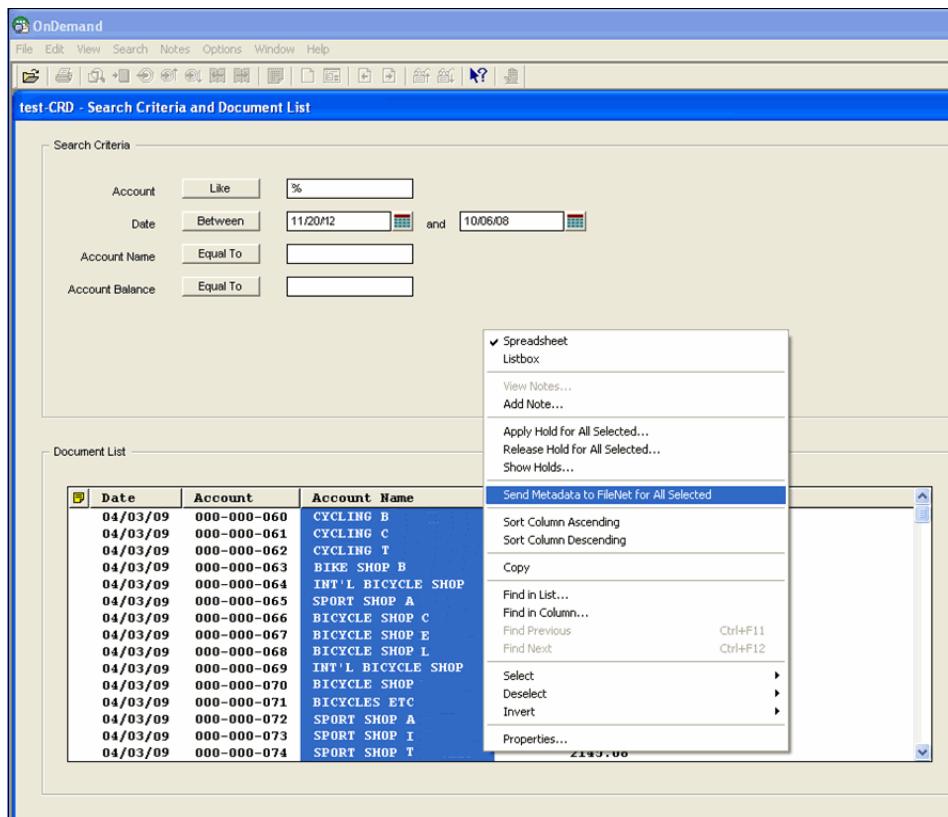


Figure 17-5 Selecting Content Manager OnDemand documents to become records

17.3 Content Federation Services for Content Manager OnDemand architecture

Content Federation Services is based on the federation implementation strategy for managing distributed content. The distinguishing feature of this strategy is the global catalog. The global catalog is a searchable database that contains information about content that is stored in various repositories and repository types in separate locations throughout the enterprise that are identified for federation. Enabling Content Federation Services for Content Manager OnDemand does not change the architectural design of Content Manager OnDemand. Content Federation Services for Content Manager OnDemand is a mid-tier process that enhances your ability to manage content more efficiently.

Figure 17-6 on page 373 shows the high-level access path to declare records from Content Manager OnDemand, Content Navigator, and Content Manager OnDemand clients.

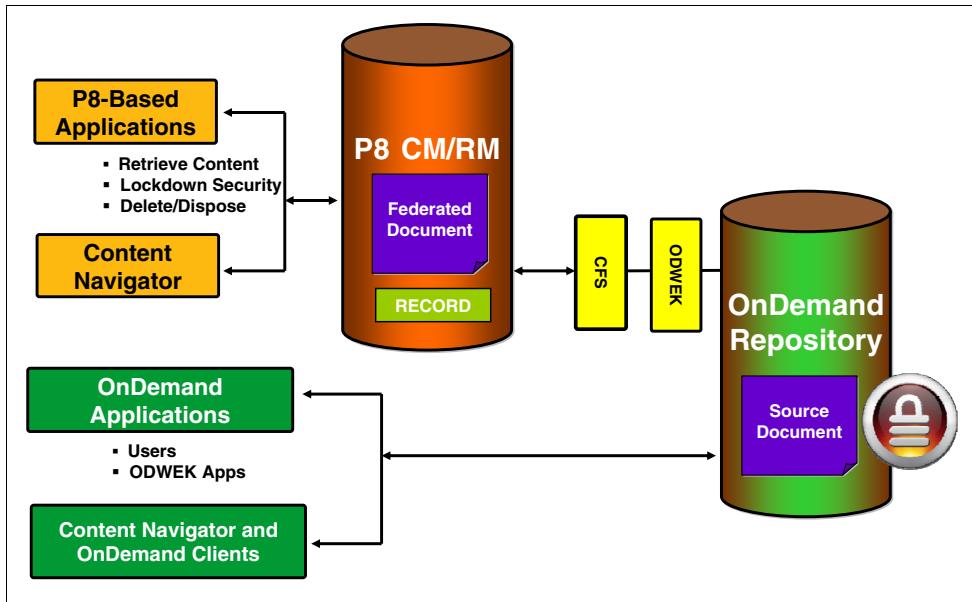


Figure 17-6 High-level access path to declared records by using FileNet Content Federation Services

17.4 Deployment considerations

Understanding the difference between Enhanced Retention Management and IBM Enterprise Records is critical to choosing the correct solution:

- ▶ Enhanced Retention Management provides for the lockdown of documents and prevents expiration.
- ▶ Enterprise Records declares documents as records that immediately become part of a corporate file plan. Enterprise Records notifies the Content Manager OnDemand repository when it reaches its event-based expiration. This action passes control to Content Manager OnDemand to delete the document unless Enhanced Retention Management is also enabled for the same document. If Enhanced Retention Management is also managing the document, it waits until the hold is released before the document is eligible for deletion.

When you use Enhanced Retention Management or Enterprise Records, Content Manager OnDemand must be in complete control of expiration processing. Therefore, if you use IBM Tivoli Storage Manager or object access method (OAM), you must disable the ability for either of these storage managers to expire data. Also, you can use Enhanced Retention Management and FileNet P8 Content Federation Services for Content Manager OnDemand only against application groups that have an expiration type of LOAD. For those application groups that have expiration types of DOCUMENT, SEGMENT, or STORAGE MANAGER, utilities are available to convert those application groups to LOAD. We recommend that you engage IBM Lab Services to provide these services.

A new parameter was added to the **arsmaint -d** and **arsadmin unload** commands. The **-D <pct_max>** parameter instructs the Content Manager OnDemand expiration process at what percentage threshold to reload the documents. For example, a value of 0 (the default) means that if any documents are held by either Enhanced Retention Management or IBM Enterprise Records, reloading never occurs. If all of the documents in a load are held, it makes no sense for Content Manager OnDemand to reload the data, so reloading never occurs. However, if the percentage of documents in the load that is held is less than the **-D <pct_max>** value, Content Manager OnDemand reloads the data, reapplies any existing holds, and deletes the original load.

In cases where both Enhanced Retention Management and IBM Enterprise Records are used, a document is not deleted until IBM Enterprise Records notifies Content Manager OnDemand that it must delete the document and all Content Manager OnDemand holds are removed from the document. With Content Manager OnDemand and FileNet P8 Content Federation Services for Content Manager OnDemand, you cannot update any metadata fields (for example, application group fields) or re-create the application that is used in federated document, and you cannot load documents that might have identical metadata. Identical metadata is highly unusually because most documents either have a date or another uniquely distinguishing value to identify them. When reloading occurs, a configuration option exists that tells Content Manager OnDemand whether to preserve any existing annotations. Keeping existing annotations incurs an additional processing impact in the reload process. The option to control annotations is in the **ars.cfg** file:

```
ARS_HOLD_CFSOD_RELOAD_ANNOTATIONS=[0 | 1]
```

This setting defaults to 0.

Various options exist for managing content to meet business challenges. Architectural design is critical to support an enterprise vision for content. We recommend that you consult with your local IBM Enterprise Content Management architects to implement the solution that meets your needs.



Part 5

Troubleshooting, hints, and techniques

This part contains the following chapter:

- ▶ Chapter 18, “Troubleshooting and tracing” on page 377



Troubleshooting and tracing

A problem can manifest itself in many different ways. Often, the root cause of the problem is not obvious. This chapter describes an approach to problem determination for the IBM Content Manager OnDemand (Content Manager OnDemand) system administrator. It guides you through the initial steps in problem diagnosis. In addition, this chapter helps you gather the documentation that is most likely required by the IBM Support Center for further diagnosis.

In this chapter, we cover the following topics:

- ▶ Troubleshooting common problems
- ▶ Information collection
- ▶ Content Manager OnDemand trace facility
- ▶ Other tracing options

18.1 Troubleshooting common problems

While administrators and users work with Content Manager OnDemand systems, they might encounter various problems. These problems can occur in several main areas. We classify them into the following categories:

- ▶ Indexing and loading issues
- ▶ Content Manager OnDemand maintenance issues
- ▶ Monitoring the main server task arssockd
- ▶ Installation and migration issues
- ▶ Common server messages

In this section, we describe several of the common problems and provide possible solutions to them. At the end of the section, we also include a list of common server messages.

Tip for determining the cause of the problems: For the UNIX platform, the console message might help determine the cause of the problem. However, if you use Telnet from your personal computer, you might miss the important console message. For AIX, you can switch the console to your current terminal by running the **swcons 'tty'** command. To switch it back to the console, run the **swcons** command.

18.1.1 Client issues

Users often encounter the following common problems when they run client-side applications:

- ▶ **Problem:** Client connection received the error Server failed while

Reasons: The following areas might cause this problem:

- Server is not up, or the server is up but not responding to a request
- TCP/IP problems between Windows and the OnDemand host
- Protocol problem
- Firewall problem

Resolution: Check the following conditions:

- OnDemand library server is up, and it accepts requests. For example, log on to the OnDemand host and issue an **arsdoc** query against the OnDemand system log.
- You can **ping** the OnDemand host. If not, consult your OS support.
- The port OnDemand server is listening and ready for a supported protocol. For example, issue the **netstat -tulpn** command.
- The Linux fire wall is not on. For example, use the **systemctl status firewalld** command to turn off the firewall by issuing the **systemctl disable firewalld** command.

- ▶ **Problem:** Content Navigator or a custom application encounters an error with OnDemand.

Resolution: Test the scenario with the OnDemand Windows client first. If the problem cannot be reproduced, turn on the OnDemand Web Enablement Kit (ODWEK) trace. (See 18.2, “Information collection” on page 389). After the ODWEK application programming interface (API) in question is identified from the trace, run a stand-alone program to test the API. (A sample program can be requested from IBM.)

18.1.2 Indexing and loading issues

The following common problems are encountered while users index or load documents:

- ▶ **Problem:** When you attempt to index a report with a large record length, you see the error message that is shown in Figure 18-1.

0425-422 AN ERROR OCCURRED WHILE ATTEMPTING TO READ /filename RETURN CODE
310.

Figure 18-1 Error message with return code 310

Reason or resolution: The maximum record length, which is 32 KB, for AFP Conversion and Indexing Facility (ACIF) might be exceeded.

Return code 310 explanation: An attempt to read a dataset failed. This message is informational and further action takes place in higher-level modules, if required. The file format is not valid.

- ▶ **Problem:** The **arsload** program performs progressively slower over time.

Reason or resolution: Performance problems can be caused by various reasons, and they require careful examination. Content Manager OnDemand issues an SQL **DELETE** against the ARSLOAD table before Content Manager OnDemand adds that same information to the ARSLOAD table to guarantee uniqueness. Duplicate information cannot be in the ARSLOAD table. This SQL **DELETE** is a single action against the ARSLOAD table and it uses an index that is formed from the application group identifier (AGID) and NAME.

The ARSLOAD table has two indexes:

- ARSLOAD_NAME_IDX, which contains the AGID and NAME
- ARSLOAD_IDX, which contains the AGID and EXP_DATE columns

Without the ARSLOAD_NAME_IDX, each load performs a complete table scan of the ARSLOAD table.

Check to see whether you already have these indexes. In addition to index creation, gather statistics by running the following command:

```
arsdb -I <INSTANCE_NAME> -mv
```

- ▶ **Problem:** The **arsload** program terminates when an unrecoverable error occurs during index, database, or storage manager processing.

Reason or resolution: Open the Content Manager OnDemand system log folder and view the messages that the **arsload** program generated during the load process. Search for message number 88 in the system log.

If the **arsload** program failed during indexing, correct the problem and then restart the load process from the beginning. Common causes of problems during indexing include invalid input files, invalid indexing parameters, invalid indexing parameter files, and insufficient temporary space.

If the failure occurred during database processing or storage manager processing, check the database or storage manager for errors.

Check the message log for a load ID that the **arsload** program saved in the system log. Before you attempt to reload the input data, you must remove the data that was created during the failed load process by using the **UNLOAD** function of the **arsadmin** program.

Restart the load process from the beginning.

- ▶ **Problem:** Content Manager OnDemand indexing fails when only one field is defined for an application group.

Reason or resolution: The Content Manager OnDemand file name indexing feature needs at minimum one index and a field value that are defined in the application indexer parameters.

Verify that you are using a file name index with one field that is defined in the application group and no field or indexing parameter defined for the application. If these conditions are true, you must use a field. You can define a dummy literal index and field value in the application indexing parameter as a placeholder. This dummy value is not processed, but it allows the file name to be indexed successfully.

- ▶ **Problem:** Content Manager OnDemand does not break up the PDF file into separate reports when TRIGGERS are defined correctly and indexing is successful. For certain reports, the trigger is not honored and the reports are grouped.

Reason or resolution: The field value *must change* for Content Manager OnDemand to indicate a report break. In Figure 18-2, there are several pages of a document. Page 1 is the TRIGGER, and the name is the field that is placed into the index.

Page 1 John Doe
Page 2 John Doe
...
...
Page 1 John Doe
...
Page 1 John Smith

Figure 18-2 Sample document for indexing

In this example, because the string Page 2 does not match the TRIGGER, it is ignored, and that page is included in report 1. Moreover, the report does not break until the name John Smith is read because it is different from the name John Doe.

- ▶ **Problem:** When the user views a document that is loaded with large object (LOB), the client receives the message:

Viewer Page count does not match Load Page Count. Viewing may be adversely affected. Contact your system administrator.

If the user clicks OK, the document can be viewed in its entirety, except that the page number is incorrect.

Reason or resolution: When a document is loaded as LOB, the Loader must count the pages, because a certain number of pages go into a LOB segment. When the client retrieves a LOB segment, the client also counts the pages that it receives from the server. The user will receive the message "Viewer Page count does not match Load Page Count" when the two page counts do not match.

This problem is usually caused by the user running ACIF to load a document that contains the form feed character 'x'0C'. ACIF does not support the form feed character as the start of a new page, but the line data viewer does support the form feed character as the start of a new page. Therefore, the viewer ends up with a different count of pages than the loader.

If you use the `asciinp` exit, ensure that the exit was not modified and recompile the exit.

If you use any exit other than the `asciinp` exit, do not use the exit so that troubleshooting will be easier.

If the file was transferred as text from a Windows system to an AIX system, the 'x'0D0A' will be changed to 'x'0A', which can affect the indexing.

Check your input file in a hex editor to verify whether the delimiter is 'x'0D0A' and that the file contains 'x'0C'.

18.1.3 Content Manager OnDemand maintenance issues

The following problems relate to Content Manager OnDemand maintenance:

- ▶ **Problem:** One of the Content Manager OnDemand database file systems is reaching 100% usage, and the file system size cannot be increased. How do you determine whether an application group is using this file system?

Reason or resolution: Follow these steps:

- a. Run the `arstblsp` command to list the open table for the application group. For example, the application group that you want to find is called AppGrpName. Use the following command:

```
arstblsp -a 3 -g AppGrpName
```

The command returns table name CAA1:

```
Table still open for loading: ApplGroup(AppGrpName) Agid(5016) Table (CAA1)
```

- b. List the table space ID, table space, and table name for the application group data table that is opened, for example:

```
su - archive  
db2 connect to archive  
db2 "select tbspaceid, tbspace, tablename from syscat.tables where  
tablename='CAA1'"
```

The command returns the following output for table space ID 3:

```
TBSpaceID TBSpace TABNAME  
3 ROOT_CAA1 CAA1
```

- c. Determine the containers for this table space ID by running the following command:

```
db2 "list tablespace containers for 3"
```

The command returns with the table space containers for table space 3, as shown in Figure 18-3 on page 382.

```

Tablespace Containers for Tablespace 3

Container ID = 0
Name = /arsdb/db1/SMS/ARCHIVE/root/CAA1.0.0
Type = Path

Container ID = 1
Name = /arsdb/db1/SMS/ARCHIVE/root/CAA1.1.0
Type = Path

Container ID = 2
Name = /arsdb/db1/SMS/ARCHIVE/root/CAA1.2.0
Type = Path

Container ID = 3
Name = /arsdb/db1/SMS/ARCHIVE/root/CAA1.3.0
Type = Path

```

Figure 18-3 Output of db2 list tablespace 3 command

- d. Check whether any of the containers that were listed previously belong to the file system that is full:

- If any of the containers that were listed previously belong to the full file system, close the opened application group data table by using the following command:

`arstblsp -a 1 -g AppGrpName`

The following message indicates that the table closed successfully:

`Closed table successfully: App1Group(AppGrpName) Agid(5016) Table(CAA1)`

- If none of the containers that were listed previously belong to the full file system, continue to find the next application group.

When the application group data table is closed, Content Manager OnDemand creates a table on a file system as defined in ARS.DBFS when data is next loaded. Content Manager OnDemand also searches for a file system with more free space to create the new table.

- **Problem:** The `arsmaint` program fails to complete.

Reason or resolution: The problem that is most commonly encountered is that the cache file system is full or a link is broken.

For a full cache file system, check to determine which file system is full, and expand the file system, if possible.

For a broken link problem, the system log displays errors that relate to `arsmaint`.

If neither situation is the case, check to see whether `arsload` is running at the same time. If `arsload` is running at the same time that you run the `arsmaint -r` command, `arsmaint` might fail.

- ▶ **Problem:** When you create new application groups, you see the error that is shown in Figure 18-4.

```
DB2 z/OS SQLCODE-497
THE MAXIMUM LIMIT OF INTERNAL IDENTIFIERS HAS BEEN EXCEEDED FOR DATABASE
<database-name>
```

Figure 18-4 Error message SQLCODE 497

Reason or resolution: The following actions can help decrease the internal limits:

- If the DBID limit is exceeded, **DROP** all unused databases and issue a **COMMIT**.
- If the OBID limit is exceeded, **DROP** all unused objects in the database and issue a **COMMIT**. Specify a different database, or run the **MODIFY** utility to reclaim unused OBIDs.
- Consider dropping indexes on application group data tables for indexes that are not frequently used for access. Review NODX reports to identify possible indexes that can be dropped.
- Find any obsolete application groups and related data whose definitions can be deleted by using the Administrator Client.
- Analyze application group data tables that became multiple segment tables, and change the MAX_ROWS column of the current table in the ARSSEG table to a larger value so that another table will not be created.
- Check your application group expiration settings. Ensure that your expiration processes are performed on a timely basis. Ensure that your expiration processes complete successfully so that application group data segment tables, table spaces, and indexes are dropped in DB2 at the same time that the expiration processing occurs.
- Consolidate applications into fewer application groups, if possible.
- Create another database and modify the **ARSUTBL** exit to change the default created table space to a new database. For more information, see Chapter 11, “Exits” on page 241.
- Use Content Manager OnDemand Administrator Client to define application groups to go to different databases.

18.1.4 Monitoring the main server task arssockd

The following problem is common:

- ▶ **Problem:** At the start, you need more information about the main server task that is named **arssockd**.

Reason or resolution: You can now monitor the main server started task **arssockd** by displaying the process usage information for the instance. By using the parameter **/-I ARC900 -x -p** for z/OS, add the following PARM statement to the **arssockd** started task:

PARM='/-I ARC900 -x -p'

-p displays the process information.

For Content Manager OnDemand, you can monitor the library server from any machine with Content Manager OnDemand installed on it by running the following command:

arssockd -I <INSTANCE> -p -x

The object servers also can be monitored from the object server by running the following command:

```
arsobjd -I <obj_hostname,port> -p -x
```

If you want to see all of the parameters that are available for **arssockd**, run **arssockd** with **-p** without any other parameters. You receive the output that is shown in Figure 18-5.

```
/usr/lpp/ars/V9ROM0/bin>arssockd -p
ARS0980I Usage: arssockd [options]
    Version: 9.0.0.1
    -h <od_inst> OnDemand instance name or host name (same as -I)
    -I <od_inst> OnDemand instance name or host name (same as -h)
    -p Display process usage information for the given instance
    -P Ping the OnDemand Instance
    -q Display configuration and version information for the given
instance
    -r <iterations> Number of iterations (defaults to 1)
    -s <seconds> Number of seconds between iterations (defaults to 1)
    -S Start the OnDemand server for the given instance
    -T Stop the OnDemand server for the given instance
    -v Verbose output
    -x Extended information (when used with -p)
    -1 <trace_file> Trace file
    -2 <trace_level> Trace level
```

Figure 18-5 Options for running arssockd with the -p parameter

18.1.5 Installation and migration issues

The following problems might be encountered when you install or migrate Content Manager OnDemand systems:

- ▶ **Problem:** Various errors occur during the installation of Content Manager OnDemand for Multiplatforms V9.5.

Reason or resolution: First, look at the *installation directory*. The new installer does not change the directory location to where the installation occurs as the installer did in the previous release. This situation might cause installation errors.

Important: Before you install the Content Manager OnDemand system, note the installation directory location because changing the directory location affects upgrade instructions. In version 9.0 and earlier releases, the installer removes the previous version of Content Manager OnDemand.

For special installation and configuration instructions, see the installation readme file.

Table 18-1 on page 385 shows the new default installation directory locations.

Table 18-1 Default installation directory

Operating system	Content Manager OnDemand server installation directory ODWEK installation directory
AIX, HP-UX, and Solaris	/opt/IBM/ondemand/V9.5 /opt/IBM/odwek/V9.5
Linux (both x64 and z Systems)	/opt/ibm/ondemand/V9.5 /opt/ibm/odwek/V9.5
Microsoft Windows	C:\Program Files\IBM\OnDemand for Windows\V9.5 C:\Program Files\IBM\OnDemand Web Enablement Kit\V9.5

- ▶ **Problem:** When you start Content Manager OnDemand, you see the error message that is shown in Figure 18-6.

```
ARS0013E ARSSOCKD DB Error: STARTUP -- SQLSTATE=1 -
ARS_ORIGINAL_CODEPAGE is not defined in ars.cfg. Run arsdb -u to
determine setting of ARS_ORIGINAL_CODEPAGE, SQLCODE=0, File=arssys.c,
Line=370
```

Figure 18-6 Error message ARSSOCKD DB error

Reason or resolution: For version 8.5.0, a new ars.cfg parameter, which is **ARS_ORIGINAL_CODEPAGE**, is *now required*.

Important: When you use Content Manager OnDemand for z/OS V8.5.0 to access a pre-version 8.5 Content Manager OnDemand database, this parameter *must* be set to the code page that the *pre-version 8.5 server* was running in. Failure to set this parameter prevents the Content Manager OnDemand server from starting. Setting this parameter incorrectly results in data corruption. This information is in the Content Manager OnDemand z/OS 8.5 readme file.

For information about the correct setting for **ARS_ORIGINAL_CODEPAGE**, run the **arsdb -u** command without **ARS_ORIGINAL_CODEPAGE** in the ars.cfg file, for example:

arsdb: Unable to initialize environment. The return code is -1 -

Define **ARS_ORIGINAL_CODEPAGE** in the ars.cfg file:

- If this instance is new, use the following setting:

ARS_ORIGINAL_CODEPAGE=37

- If this instance is an existing instance, use the following setting:

ARS_ORIGINAL_CODEPAGE=1047

Important: After you define **ARS_ORIGINAL_CODEPAGE**, *you must never change it*.

For an existing instance (created before version 8.5), the code page that is displayed must match the code page in the ARS0220I message that is issued by the pre-version 8.5 server when it starts, regardless of the setting of **ARS_ORIGINAL_CODEPAGE**. See this example message:

ARS0220I Server code page is 1047

Note: An 8.5 server will always display

ARS0220I Server code page is 1200

- **Problem:** You encounter an error while **arsload** is running, as shown in Figure 18-7.

ARSLOAD Command: An error occurred. Contact your system administrator and/or consult the System Log. File=arsadmp.c, Line=1608 Failed while attempting to load the database. The last row successfully loaded was 117461. Loaded 117461 rows into the database

Figure 18-7 ARSLOAD error message

Reason or resolution: This issue might relate to an incorrect **ARS_ORIGINAL_CODEPAGE** setting. Check that the value is correct by using the following method. For more information, see Technote 1616768.

- Use this method for UNIX servers:

For information about the correct setting for **ARS_ORIGINAL_CODEPAGE**, run **arsdb -u** without **ARS_ORIGINAL_CODEPAGE** in the **ars.cfg** file, for example:

```
arsdb -u -I <OD_INSTANCE>
```

You might receive this message:

```
arsdb: Unable to initialize environment. The return code is -1.
```

Define **ARS_ORIGINAL_CODEPAGE** in the **ars.cfg** file:

- If this instance is a new (created in version 8.5) instance, use the following setting:
ARS_ORIGINAL_CODEPAGE=819
- If this instance is an existing instance (created before version 8.5), use the following setting:
ARS_ORIGINAL_CODEPAGE=923

After it is set, **ARS_ORIGINAL_CODEPAGE must never change**.

Edit the **ars.cfg** file and add the **ARS_ORIGINAL_CODEPAGE** parameter, which is set to the appropriate value by the **arsdb** command.

- Use this method for Windows servers:

For information about the correct setting for **ARS_ORIGINAL_CODEPAGE**, run **arsdb.exe -u -I <OD_INSTANCE>**, for example:

```
arsdb.exe -I <OD_INSTANCE> -u
```

You might receive the following output:

```
arsdb: Unable to initialize environment. The return code is -1.
```

Define **ARS_ORIGINAL_CODEPAGE** in the **ars.cfg** file:

- If this instance is a new (created in version 8.5) instance, set **ARS_ORIGINAL_CODEPAGE** with a value of 1208 in the registry.
- If this instance is an existing instance (created before version 8.5), set **ARS_ORIGINAL_CODEPAGE** with a value of 5348 in the registry.

The registry setting must be in the **HKEY_LOCAL_MACHINE\SOFTWARE\IBM\OnDemand for Windows\@SRV@_<OD_INSTANCE>\CFG** registry key.

After it is set, **ARS_ORIGINAL_CODEPAGE must never change**.

Run **regedit.exe** and update the Windows registry key that is specified in the output of the **arsdb** command and add the **ARS_ORIGINAL_CODEPAGE** string value, which is set to the appropriate value by the **arsdb** command.

- ▶ **Problem:** When you run the `arsdb -I ARCHIVE -vu` command from a BPXBATCH job, you see errors in STDOUT or in the output of `arsdb -c`, as shown in Figure 18-8.

```
DB Error: {DB2 for OS/390}{ODBC DRIVER} SQLSTATE=58004 ERRLOC=2:170:9
CAF "CONNECT" failed using DB2 system:DB2K
RC=08 and REASON=00f30002 -- SQLSTATE=58004, SQLCODE=-99999
arsdb:.Unable to connect to DB2 ARSDBASE database
or
arsdb: Unable to determine the database engine
```

Figure 18-8 ARSDB error

Reason or resolution: Enter an `export` command for DSNA0INI in the BPXBATCH job, or ensure that `export DSNA0INI='/etc/ars/cli.ini'` was defined. Verify that your `cli.ini` file is in this directory. Also, check that your `cli.ini` file references the correct DB2 subsystem, for example:

```
[COMMON]
MVSDEFAULTSSID=DB1M
[DB1M]
PLANNAME=DSNACLI
```

- ▶ **Problem:** During installation and migration, when you run `arsdb -u -I` or `arsdb -I ARCHIVE -vx ARSSYS`, you receive an error similar to Figure 18-9.

```
arsdb -u -I
CEE3204S The system detected a protection exception (System Completion
Code=0C4).
From entry point u_file_write_44_arsxh at compile unit offset
+00000054 at entry offset +00000054 at address
```

Figure 18-9 ARSDB error during installation and migration

Reason or resolution: Multiple resolutions can be attempted when you encounter this error:

- Check the `ARSSOCKD` region parameter. We recommend that you use `region=0`. Also, check your TSO logon region size. Increase TSO, log out of TSO, and log back in.
- When you run `export (x)`, the contents of the ARSxxx table are exported to a flat file. The command attempts to write the file to the directory in which the `arsdb` command runs. Ensure that the user has write permissions for the file. Also, ensure that the user has permissions for the `ARS_TMP` intermediate file.
- Check the OMVS size to see whether it can be increased to the system limit (`D OMVS,L`). Check your OMVS limits: `MAXMMAPAREA` and `MAXSHAREPAGES`. The following example is output of the `D OMVS,L` command:

	CURRENT USAGE	HIGHWATER USAGE	SYSTEM LIMIT
MAXMMAPAREA	3107	3107	4096

The example shows current `MAXMMAPAREA` is 3107 and the maximum size that `MAXMMAPAREA` can go is 4096. In our experience, version 8.5 required at least 3646 for the value of `MAXMMAPAREA`. In this example, the minimum memory size needed for CMOD will be $3107 + 3646 = 6753$. Therefore, you must increase the `MAXMMAPAREA` from 4096 to a higher value, for example, 40960, to solve the problem.

- ▶ **Problem:** When you run `/usr/lpp/ars/V9R0M0/bin/arsdb -I ARCHIVE -iv arsag`, you receive an error message similar to Figure 18-10.

```
unable to import table arsag. err=1904
```

Figure 18-10 ARSDB error message 1904

Reason or resolution: The `arsdb` command cannot open a temporary file for messages. `ARS_TMP` from the `ARCHIVE` instance is not being used. If `ARS_TMP` is not present, root '/' is used. To resolve this error, define the directories correctly so that they are pointed to by the `ARS_TMP=` parm in the `ars.cfg` file.

18.1.6 Common server messages

Several common messages can occur in a Content Manager OnDemand environment:

- ▶ **ARS0066I message**

```
ARS0066I Application Group Document Get: Name(appl_grp_name) Agid(agrid)
ApplName(appl_name) Aid(aid)NodeName(node_name) Nid(nid) Server(server)
Time(time) Fls(fields)
```

This message is received during document retrieval from a specific application group. You can find this message in the Content Manager OnDemand system log. The message is for your information only.

This message is valuable because it records the document that was retrieved and other information about the document and the retrieval time, for example:

```
ApplGroup DocGet: Name(QPJOBLOG) Agid(5081)
ApplName(QPJOBLOG)Aid(5082)NodeName(-CACHE-)Nid(1)Server(-LOCAL-) Time(0.322)
Fls()
```

- ▶ **ARS0067I message**

```
ARS0067I Application Group Resource Get: Name(appl_grp_name) Agid(agrid)
NodeName(node_name) Nid(nid)Server(server) Time(time)
```

This message is received when a resource is retrieved from a specific application group. You can find this message in the Content Manager OnDemand system log. The message is for your information only.

This message is valuable because it records the name of the application group that the resource is associated with and the time of the resource retrieval, for example:

```
ApplGroup ResGet: Name(INS) Agid(6843) NodeName(-CACHE-) Nid(25)
Server(-LOCAL-) Time(0.069)
```

- ▶ **ARS0087I message**

```
ARS0087I Application Group Load: Name(appl_grp_name) LoadId(load_id) File(file)
InputSize(input_size)OutputSize(output_size)
```

This message is received when `arsload` is running. A report is loaded into the system. The message identifies the application group, the input file, and the load ID. This message is for your information only.

This message is valuable because it records information about the load, such as the application group name, load ID, file name, and sizes of the files at load time, for example:

```
ApplGroup Load: Name(MOSUNPO)LoadId(5535-2-0-1FAA-12349-12349)
File(/QIBM/USERDATA/ONDEMAND/QUSROND/TMP/SP_MOSUNPO_WTH7HTWCXA_DBRYANT_064315_0
00009_RDR400M_1031023_210136) InputSize(225789) OutputSize(16380)
```

- ▶ ARS0088E message

```
ARS0088E Application Group Failed Load: Name(appl_grp_name) LoadId(load_id)
File(file)
```

This message is received when the load process fails. You can find this message in the Content Manager OnDemand system log.

This message is valuable because it records the name of the application group, load-id, and file name of the failed load, for example:

```
ApplGroup Failed Load: Name(LATECHARGE) LoadId()
File(/QIBM/USERDATA/ONDEMAND/QUSROND/TMP/SP_QPRLR133_QPRTJOB_DBRYANT_001467_000
022_RDR400M_1021226_132052)
```

Response: To correct the problem, see the other messages that were generated by the **ARSLLOAD** program and see the messages in the Content Manager OnDemand system log. Then, resubmit the command.

18.2 Information collection

If the guidance in 18.1, “Troubleshooting common problems” on page 378 does not help you determine and resolve your problem, speak to IBM Support. In this section, we explain the information to gather for IBM Support so that they can help you more efficiently.

When you report a problem to IBM Support, you must provide the version of the software that you are using. For Content Manager OnDemand, this version might include the numbers of the operating system, DB2, Oracle, IBM Tivoli Storage Manager, Content Manager OnDemand, and ODWEK. This information helps IBM Support determine whether the software version is still supported and whether known issues exist with that software version.

We also advise that you apply the latest maintenance level to Content Manager OnDemand before you contact IBM Support to ensure that you are not experiencing a problem that is resolved.

Table 18-2 shows commands that are used to determine the version of Content Manager OnDemand on various operating systems.

Table 18-2 Determine the version of Content Manager OnDemand

Operating system	Example of the command to determine the version
IBM AIX	<code>1s1pp -l grep ars</code> (OnDemand server and ODWEK)
Sun Solaris	<code>pkginfo -l ondemand</code>
HP-UX	<code>swlist -l product grep</code> (OnDemand server and ODWEK)
Linux	Look for the highest version for the package name in the list: <code>rpm -qa grep ondemand</code> (Take highest version that is listed.)
Windows	From the Content Manager OnDemand configurator, click Help → About .
Windows client	From the Windows client, click Help → About OnDemand .
ODWEK	Check the logon message.
Content Manager OnDemand commands	Starting in Content Manager OnDemand 8.5, the response to all Content Manager OnDemand commands includes the release and fix pack level.

After you obtain the correct version number of the software that you are using, you must collect information that is specific to the problem.

Problems can occur in several main areas. We divide them into the following areas in this section:

- ▶ Indexing or loading
- ▶ Database
- ▶ Tivoli Storage Manager
- ▶ Content Manager OnDemand Client logon
- ▶ Performance
- ▶ ODWEK
- ▶ Content Manager OnDemand server hangs or crashes

In 18.2.8, “Exporting information to a local server” on page 397, we demonstrate how to export Content Manager OnDemand information, such as an application group, application, and folder, to a local server.

18.2.1 Indexing or loading

This section describes the logs to collect that relate to indexing or a loading problem.

Common loading issues

Table 18-3 shows the information to collect if a problem occurs with loading.

Table 18-3 Information to collect for loading issues

File	Description
ARSSOCKD.ERR	This log file is for the arssockd daemon process. The process is instance-dependent if multiple instances are running.
ARSLOAD error message	The ARSLOAD error message shows whether ARSLOAD failed at the indexing or loading phase. (See ARSLOAD common error messages in 18.1.6, “Common server messages” on page 388.)
ARS.INI	This file is the Content Manager OnDemand instance configuration file. Each instance has a section in the ARS.INI file.
Content Manager OnDemand system log	This Content Manager OnDemand system log is in the system log folder. Various message numbers about warnings or errors at the time of failure are included.
Export folder, application group, and application files and sample data	The export files are used to import to the test server for problem replication.
CORE	This file holds the core memory dump that is generated by the operating system.
Version or level of DB2, Oracle, or SQL Server and Content Manager OnDemand	This file name contains the version or level of software that the server is using. Sometimes, a problem might be resolved by upgrading to the latest program temporary fix (PTF) or maintenance level.

IBM Content Manager OnDemand - Messages and Codes, SC19-3356, describes the error message codes that are in the Content Manager OnDemand system log.

Common AFP indexing problems

Content Manager OnDemand cannot load AFP data without indexes; therefore, you must first ensure that your AFP data is already indexed. Therefore, AFP must have Tag Logical Elements (TLEs).

Table 18-4 shows the information to collect when you have problems with AFP.

Table 18-4 Information to collect for common AFP problems

File	Description
Export folder, application group, and application files and sample data	The export files are imported to the test server for problem replication.
ACIF indexer error message	This file contains the error messages that are generated by the ACIF indexer. In z/OS, this file can be the job log, which has the indexer information from the failed job.
AFP sample data file	This file is a non-confidential data file that can be viewed by the IBM Support team to verify the AFP syntax.
AFP interim files that are used by AFP viewer within Content Manager OnDemand Windows Client	These files are created in the user's temporary directory. They are deleted automatically after the document is closed by AFP viewer. They are useful in determining whether the issue is a server or client issue. In the Windows client, click File → Show Temporary File Locations to see the names of the directories where the client stores the data and resource files.
AFP trace report	AFP viewer trace can be turned on by modifying the FTDPRT2.INI file in the Content Manager OnDemand installation directory. For Windows 7, the default path is shown: C:\Program Files (x86)\IBM\OnDemand Clients\V9.5\bin
AFP resource and font files	Sometimes, this file is useful for various AFP issues, such as overlay, company logo, or globalized fonts.

Before you log a problem with IBM Support, use the information in Table 18-4 to look for clues about your problem. You can check the error codes from the ACIF indexer in *IBM Content Manager OnDemand - Messages and Codes*, SC19-3356. You might find the solution immediately. If you have an AFP memory dump tool, you can also dump the AFP data file to check for an invalid AFP data stream, which is a common problem.

Note: Because the AFP data stream can be printed by an AFP printer, it does not necessarily have the correct AFP structure for loading into Content Manager OnDemand. The loading of AFP data requires a more specific AFP structure than printing AFP data. *IBM Content Manager OnDemand for Multiplatforms - Indexing*, SC19-3354, provides information about the correct AFP data stream structure.

18.2.2 Database

For DB2 problems, collect the information in Table 18-5 on page 392 for problem determination.

Table 18-5 Information to collect for DB2

File	Description
db2diag.log	This file is the DB2 diagnosis file. It is in the <code>\$HOME/sqllib/db2dump</code> directory, where <code>\$HOME</code> is the home directory of the DB2 instance.
CLI trace (Open Database Connectivity (ODBC))	This file contains the call-level interface (CLI) trace file for diagnosing SQL statements. The CLI trace option must be turned on to collect the file. (See “Setting the CLI trace for DB2” on page 392.)
Text summary	Collect a text summary of the application group and folder with the database problem by logging on to the Content Manager OnDemand server through the Content Manager OnDemand Administrator Client.
Content Manager OnDemand system log	Gather the Content Manager OnDemand system log messages 10 minutes before and after the database problem.
SQLCODE error message	If the SQLCODE error message is available, collect this information to determine whether the problem is from Content Manager OnDemand or the database.
DB2 configuration report of the Content Manager OnDemand instance	This report is generated by the <code>db2</code> command: <code>db2 get db cfg for instance_name</code>
ars.ini	If you cannot determine the database engine, check the path of the <code>ars.ini</code> file. In UNIX System Services, run <code>set > /tmp/set.txt</code> and send a binary copy of the <code>ars.ini</code> file to IBM Support.
ars.cfg	Send a binary copy of the <code>ars.cfg</code> file to IBM Support.
cli.ini	Verify the contents of the <code>cli.ini</code> file. Verify that <code>MVSDEFAULTSSID=xxxx</code> points to the correct DB2 Subsystem ID. Also, check whether the value can be set within your installation by a <code>DSAN0INI DD</code> statement in JCL or as a hierarchical file system (HFS) file.
On Windows and other platforms	For Windows, collect the Content Manager OnDemand server configuration settings: <code>HKEY_LOCAL_MACHINE\SOFTWARE\IBM\OnDemand</code> For all other platforms, collect the <code>ars.ini</code> , <code>ars.cfg</code> , <code>ars.cache</code> , and <code>ars.dbfs</code> files.
Application group report	The application group ID is the name of the related DB2 tables.

Setting the CLI trace for DB2

We list two methods to turn on the CLI trace for DB2. One method is to edit the `db2cli.ini` file directly. The other method is to use the DB2 command line.

The examples show the common options for the DB2 CLI trace. The IBM Support team might suggest a different option to collect information that is appropriate to your situation. Modify these options as advised.

In both cases, the trace file that is collected (as shown in Example 18-1 on page 393 and Example 18-2 on page 393) is in the `/tmp/db2trace.dmp` file.

Method 1: Setting up the trace by editing the db2cli.ini file

You can set up trace by editing the db2cli.ini file by completing the following steps:

1. Add a section that is similar to the section that is shown in Example 18-1 or Example 18-2 to the db2cli.ini file, depending on your platform.

For Windows, this file is in the \sql1ib path, for example, C:\Program Files\IBM\SQLLIB.

For UNIX, this file is placed in the /sql1ib/cfg path of the home directory of the instance owner, such as /home/archive/sql1ib/cfg. For z/OS, this file is in the UNIX System Services /tmp file.

Example 18-1 Common section of the db2cli.ini file

```
[COMMON]
TRACE=1
TRACEREFRESHINTERVAL=5
TRACEFILENAME=/tmp/db2trace.dmp
TRACEFLUSH=1
TRACECOMM=1
```

Example 18-2 Common section of the cli.ini file for DB2 z/OS

```
[COMMON]
MVSDEFAULTSSID=DB1X
APPLTRACEFILENAME=/tmp/db2trace
APPLTRACE=1
TRACETIMESTAMP=3
[DB1X]
PLANNNAME=DSNACLI
```

For z/OS, controls also exist for the diagnostic trace:

```
DIAGTRACE=1
DIAGTRACE_BUFFER_SIZE=6291456
DIAGTRACE_NO_WRAP=1
```

The full path of the TRACEFILENAME must be a valid directory with permission for everyone to write.

For z/OS, ensure that you refer to the following file:

```
//DSNAOINI DD PATH='/usr/lpp/ars/V9R0M0/config/cli.ini'
```

2. Restart the application (in this case **arssockd**) for the changes to take effect.
3. Re-create the DB2 problem and capture the trace information.
4. To turn off the trace, modify the db2cli.ini file again and set TRACE=0.
5. Restart **arssockd**.

Method 2: Setting up the trace by using the DB2 command line

Alternatively, you can use the DB2 command line to activate the trace by completing the following steps:

1. In the DB2 instance, run the DB2 commands that are shown in Example 18-3.

Example 18-3 Turning on the trace through the DB2 command line

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 1  
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceRefreshInterval 5  
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceFileName /tmp/db2trace.dmp  
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceComm 1  
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceFlush 1
```

2. Restart the application (in this case **arssockd**) for the changes to take effect.
3. Re-create the DB2 problem and capture the trace information.
4. Run the following command to turn off the traces:

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 0
```

5. Restart **arssockd**.

18.2.3 Tivoli Storage Manager

For Content Manager OnDemand problems that relate to Tivoli Storage Manager, collect the information that is shown in Table 18-6. For specific Tivoli Storage Manager errors, see *Collecting Data: Read First for Tivoli Storage Manager Products*, reference number 1263547.

Table 18-6 Information to collect for Tivoli Storage Manager

File	Description
Application group report	The summary information for storage management shows the storage set name, which relates to Tivoli Storage Manager.
Storage set report	This information provides the node name at Tivoli Storage Manager.
Tivoli Storage Manager activity log	This log shows the events in the Tivoli Storage Manager server. You can retrieve the log by running the Query actlog command.
Tivoli Storage Manager error message	Tivoli Storage Manager error messages are prefixed with ANS, ANR, and so on. This error is generated by Tivoli Storage Manager and can be used for Tivoli Storage Manager support for further diagnosis.

You can gather the various object reports, such as the application group report and storage set report, by right-clicking the object and selecting **Summarize**.

18.2.4 Content Manager OnDemand Client logon

If a Content Manager OnDemand Client fails to log on to the server, check that **arssockd** is running on the server. Then, check the network connectivity by performing a **ping** test from the command window of the client. Open the command window and **ping** the host name or the IP address of the Content Manager OnDemand server.

Collect the files that are listed in Table 18-7 on page 395 for client problems, such as logging in to Content Manager OnDemand.

Table 18-7 Information to collect for client logon problems

File	Description
ARS.INI	This file is the Content Manager OnDemand instance configuration file. The instance is configured in each section in the ARS.INI file.
ARS.CFG	This file is the Content Manager OnDemand configuration file.
ARSSOCKD.ERR	This file is the log file for the arssockd daemon process. The process is instance-dependent if multiple instances are running. This file is in the path that is defined for ARS_TMP .
Content Manager OnDemand system log	Check the Content Manager OnDemand system log for any specific messages that relate to logging in to the client.
Print screen	Print a screen capture of any client errors you might receive when you log on to the client for further analysis.

18.2.5 Performance

For Content Manager OnDemand performance issues, gather the information that is shown in Table 18-7.

Table 18-8 Information to collect for performance issues

File	Description
Application group report	Check these fields in the report, whether they are indexed or filters. Simply reviewing this report might resolve the issue.
Text summary	Collect a text summary of the application group and folder with the performance problem by logging on to the Content Manager OnDemand server through the Content Manager OnDemand Administrator Client.
CLI trace (ODBC)	This file contains the call-level interface (CLI) trace file for diagnosing SQL statements. The CLI trace option must be turned on to collect the file.
Content Manager OnDemand system log	Gather the Content Manager OnDemand system log messages 10 minutes before and after the search problem.
On Windows and other platforms	For Windows, collect the Content Manager OnDemand server configuration settings from the following registry key: <code>HKEY_LOCAL_MACHINE\SOFTWARE\IBM\OnDemand</code> For all other platforms, collect the following files: <code>ars.ini</code> , <code>ars.cfg</code> , <code>ars.cache</code> , and <code>ars.dbfs</code>
Database reorganization information	This file is used to check whether the arsdb command ran to reorganize -m for DB2 and SQL Server, run maintenance on the Content Manager OnDemand database, and reorganize the Content Manager OnDemand system tables. This option refreshes the tables and optimizes access to information in the database. You might also want to run the reorg command on data tables. After you reorganize the tables, run the runstats command on the tables arsdb -s , and run database statistics.
Memory information	This file contains the amount of physical memory and the memory setting in the server, such as the output from the ulimit command.
ARSSOCKD.ERR	This log file is for the arssockd daemon process. The process is instance-dependent if multiple instances are running. This file is in the path that is defined for ARS_TMP .

File	Description
Indexer information from the application report	This file helps you determine whether the report has a single index, which uses up memory if the report is huge. Also, if a large report is not using the large object option, a user might experience a long download time.
IBM Tivoli OMEGAMON® XE for DB2 Performance Monitor on z/OS	If necessary, you are instructed by IBM Support to run the Performance Monitor to further analyze a performance problem. The initial setup includes accounting and statistics reports.

18.2.6 ODWEK

For ODWEK problems, gather the information that is shown in Table 18-9. Depending on the environment and the specific failure, part of the information might not be present in your environment. See *MustGather: ODWEK Java API terminating without warning*:

<http://www.ibm.com/support/docview.wss?uid=swg21243419>

Table 18-9 Information to collect for ODWEK

File	Description
ODWEK trace	This file is the arswww.trace file.
IBM WebSphere Application Server	WebSphere MustGather crash files.
HTTPd log	This file is the IBM HTTP Server log file.
Test case	This file is a skeleton test case that can re-create the crash.
Core	This core file is generated by the operating system.
Screen captures of the problem	Provide a file that contains screen captures of the error message or document. This file is also useful for non-English error messages and documents.
Plug-ins or applets information	Provide a file of any plug-ins or applets that are used. This information helps to verify the version in use. Sometimes, a problem is resolved by using the latest version.
Version or level of ODWEK, DB2, Oracle, or SQL Server, and Content Manager OnDemand	Provide the version or level of software that the server is using. Certain issues might be resolved by simply upgrading to the latest PTF.

18.2.7 Content Manager OnDemand server hangs or crashes

For problems where Content Manager OnDemand server hangs or crashes, you can search for a few MustGather technotes by going to the following website:

<http://www.ibm.com/software/data/ondemand/mp/support.html>

Search this website by using the MustGather keyword to find the following technotes:

- ▶ *MustGather: Content Manager OnDemand Server for Windows - Hang or performance degradation*, reference # 1223907
- ▶ *MustGather: Content Manager OnDemand Server for Windows - Crash*, reference # 1226443

- ▶ *MustGather: IBM Content Manager OnDemand server hang or performance degradation on AIX*, reference # 1222374
- ▶ *MustGather: IBM Content Manager OnDemand server crash on AIX*, reference # 1223109

Follow the instructions from the technotes to gather information when the server hangs or crashes.

18.2.8 Exporting information to a local server

IBM Support might require information about the Content Manager OnDemand application group, application, and folder for problem determination.

To create a local server to export object information, complete the following steps:

1. Create a local server on your workstation:
 - a. From your Content Manager OnDemand Administrator Client, select **OnDemand Servers** and then click **File** → **New Server**. See Figure 18-11.

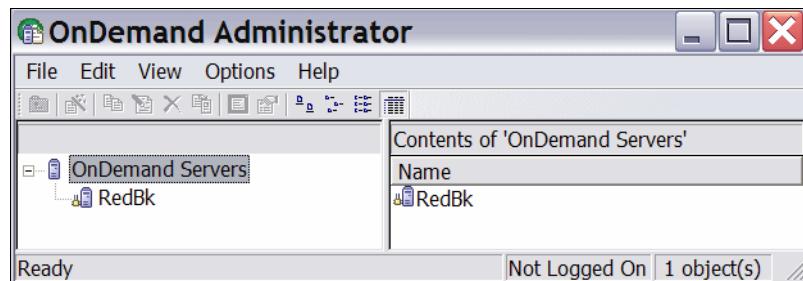


Figure 18-11 Setting up a local server

- b. From the Add a Server window that opens, for the Protocol field, select **Local**, and enter the information that is shown in Figure 18-12. Click **OK**. A local server with the name **ODlocal** is created.

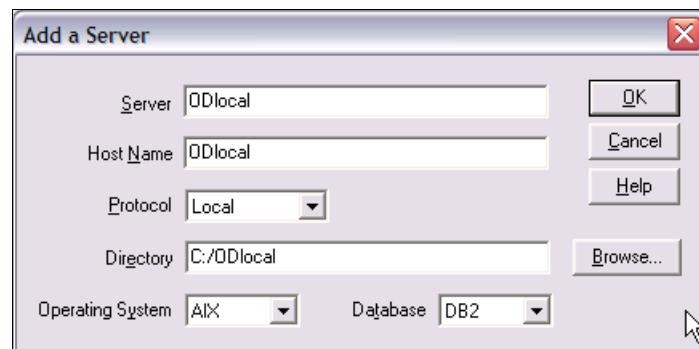


Figure 18-12 Add a Server window

2. The local server cannot be used until it is set up. Right-click the **ODlocal** server and select **Setup**, as shown in Figure 18-13 on page 398.

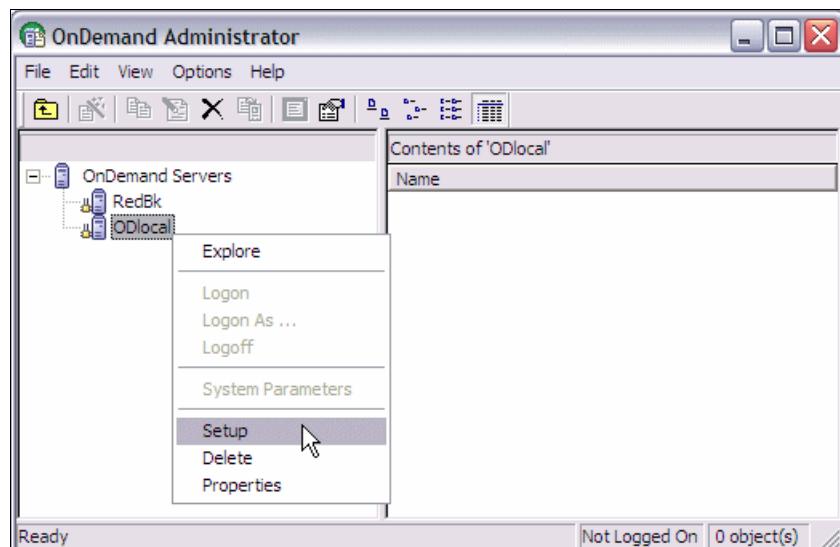


Figure 18-13 Setting up the local server

When you see the prompt “Are you sure?”, click **OK**.

When the setup is complete, the local server is ready to use. By default, the local server has a user that is named admin without any password.

3. Export the requested information from your server to the local server. Right-click the object and select **Export**. For example, if you want to export the application group with the name Redbk, right-click the object **Redbk** and select **Export**, as shown in Figure 18-14.

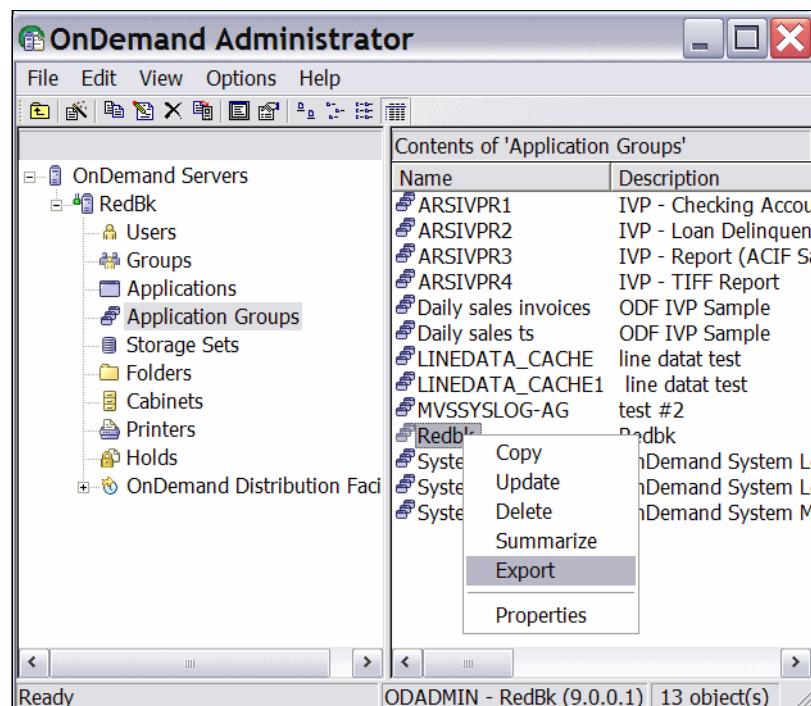


Figure 18-14 Export Application Groups

4. In the Export Application Groups window (Figure 18-15) that opens, export your application groups by completing the following steps:
 - a. From the Server list, select the server to be exported.
 - b. Click **Export**. The information of the application group that you chose starts transferring to ODlocal.
 - c. Check the message at the end of the export to ensure that the export is successful.
 - d. You can select either of the following options:
 - Select **Ignore Warnings** if you want Content Manager OnDemand to add an item regardless of any warnings. Otherwise, Content Manager OnDemand stops transferring the item when the first warning is encountered. For example, if the application group has users and groups permissions that are defined in the source server, but the users and groups are not present in the local server, the export fails. If the item to be exported exists on the destination server, the export also fails.
 - Select **No Storage Set** if you do not want Content Manager OnDemand to assign a storage set to the application group.

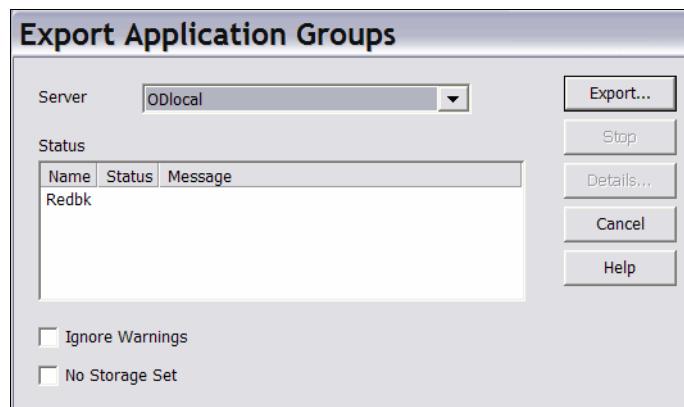


Figure 18-15 Export local server

5. When all of the requested information is exported to the local server, compress the entire directory from the directory of the local server. In this example, the directory of the local server is C:\ODlocal, as shown in Figure 18-12 on page 397.

Tip: When you export all of the requested information, we recommend this order:

- ▶ Printers
- ▶ Users and groups
- ▶ Storage sets
- ▶ Application groups, applications, and folders

18.3 Content Manager OnDemand trace facility

Content Manager OnDemand incorporated a trace facility to help IBM Support perform problem determination. In this section, we show you how to enable trace. The trace affects Content Manager OnDemand server performance. Enable the trace only when this action is requested by IBM Support. The trace is enabled to gather documentation and it must be disabled afterward.

Note: The information from the Content Manager OnDemand trace facility is for informational purposes only and must be used only under the direction of IBM Support personnel. How tracing is enabled and what gets traced is subject to change at any time. Do not use trace as a programming interface.

18.3.1 Enabling the trace facility

This information is also covered in the Technote *How to enable trace in Content Manager OnDemand server*, 1330810.

To enable the server trace facility, complete the following steps:

1. Locate your `trace.settings` file:
 - On AIX, it is in the `/opt/IBM/ondemand/V9.5/config` directory.
 - On Sun, it is in the `/opt/IBM/ondemand/V9.5/config` directory.
 - On Windows, it is in the `C:\Program Files\IBM\OnDemand Server\V9.5\config` directory.
 - On z/OS, it is in the `/SYSTEM/etc/ars` directory.
 - On other platforms, it is in the `/opt/ondemand/v9.5/config` directory.
2. Edit the `trace.settings` file to trace server startup routines by setting the following trace parameter:
`TRACE_LEVELS=ALL=15`
3. Edit your `ars.cfg` file by adding the `ARS_TRACE_SETTINGS` parameter and referencing the full path to your `trace.settings` file:
`ARS_TRACE_SETTINGS=/usr/lpp/ars/V9R0M0/config/trace.settings`

For Windows, the `ars.cfg` configuration is in the registry. Edit the registry settings for the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\IBM\OnDemand` for `WinNT\@SRV@_ARCHIVE\CFG`, where `ARCHIVE` is the name of your Content Manager OnDemand instance. Create a string value `ARS_TRACE_SETTINGS` and set the value to the full path to your `trace.settings` file.

After you enable tracing, start the Content Manager OnDemand Administrator Client.

18.3.2 Setting trace parameters

After you enable tracing, you can set the appropriate option for a runtime trace by using the Content Manager OnDemand Administrator Client.

Log on to the Content Manager OnDemand Administrator Client and configure tracing by completing the following steps:

1. Right-click the server name and select **Trace Parameters**, as shown in Figure 18-16.

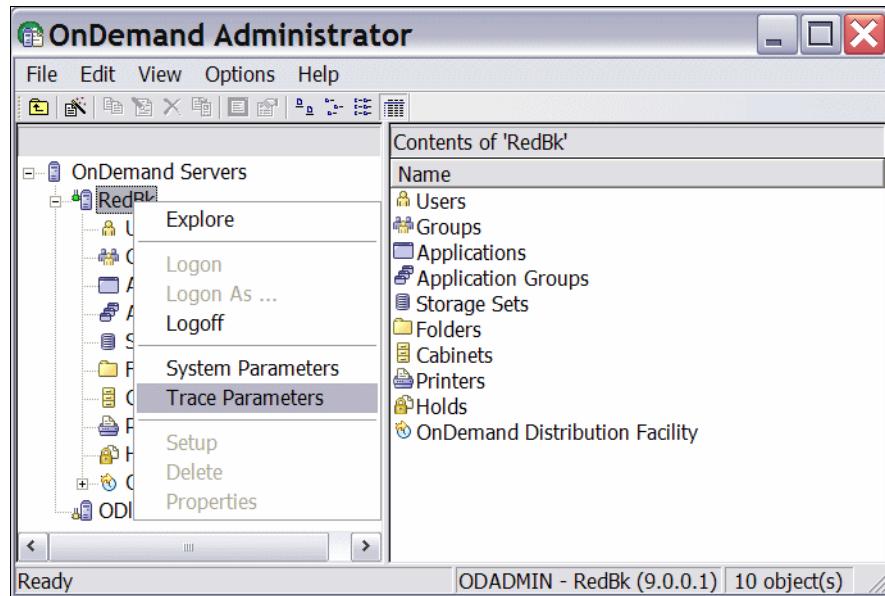


Figure 18-16 Configure trace parameters

2. In the System Trace Setting window (Figure 18-17), complete the following steps:
 - a. Select the **Activate System Trace** check box to turn on tracing for the whole system.
 - b. Enter information in the Trace Parameters entry field. The trace parameters can be name=value pairs that are separated by commas to define the trace level. These name=value pairs are provided by IBM Support. For an example, see Figure 18-17.

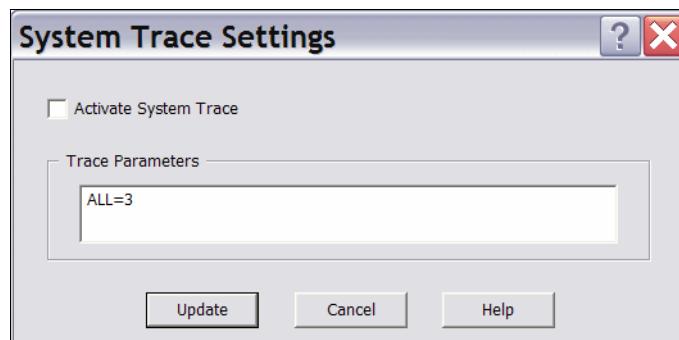


Figure 18-17 System trace settings

- c. Click **Update**. You do not need to restart Content Manager OnDemand.

After the trace is collected, you can send the trace file to IBM Support.

Note: You can stop or start the runtime trace from the Content Manager OnDemand Administrator Client anytime without restarting `arssockd`.

Important: Use trace only with the help of IBM Support because activating trace might severely affect the performance of the Content Manager OnDemand system.

18.4 Other tracing options

In addition to Content Manager OnDemand tracing, you can run other traces for additional diagnosis, as needed. Other traces are enabled to focus on specific areas and gather the necessary documentation. You need to disable the other traces afterward.

18.4.1 ARSLOAD

The **ARSLOAD** program is the main Content Manager OnDemand data loading and indexing program where you can trace data loading and indexing issues.

In most cases, trace parameters can be specified on the command line when you run **arsload**.

With the **ARSLOAD** command, the trace is enabled by the **-1** and **-2** parameters:

- ▶ **-1 <trace_file>** (fully qualified trace file name)
- ▶ **-2 <level>** (trace level number)
- ▶ The trace level number values are additive (default is 3):
 - 1: Errors
 - 2: Warnings
 - 4: Info
 - 8: Flow

These trace levels provide entry and exit information for functions.

The trace level numbers are added up, and the default level is 3, which is used to report errors and warnings that occur during loading.

You can also use name=value pairs that are separated by commas to define the trace level. The name=value pairs are provided by IBM Support when they request the trace, for example:

```
-1 trace.out -2 ALL=15,ARSRD=3,ARSLOAD=3
```

The trace no longer generates textual output, so it performs better. The output is now in binary format and requires the use of the **arstfmt** command, which is in the **bin** directory of your Content Manager OnDemand server (version 8.5 and later). You can format the output in either text or XML format.

To produce a text formatted trace file, run the following command:

```
/usr/lpp/ars/V9R0M0/bin/arstfmt -i /tmp/arswww.trace -o /tmp/arswww.trace.txt
```

To produce an XML formatted trace file, run the following command:

```
/usr/lpp/ars/V9R0M0/bin/arstfmt -x -i /tmp/arswww.trace -o /tmp/arswww.trace.xml
```

Limitation: Use the **-1** and **-2** parameters only under the supervision of IBM Support because they might affect performance.

18.4.2 MidServer trace (z/OS only)

To collect data that processed by the MidServer, complete the following steps:

1. Locate the MidServer arsMSVR.cfg configuration file. The file is in the following directory:
/MountPoint/config/midserver
2. Turn on MidServer tracing by setting MIDSERVERTRACE=1.

To collect the input data that is returned to the application, set traceLevel to 2 before you issue the logon function request. This traceLevel indicates that a full trace by the C stub is requested.

3. Run the application to re-create the problem.
4. Send the following files to IBM Support for further analysis. All of these files that are sent must be from the same test.

File name	Location
arswww.ini	/MountPoint/config/midserver
arswww.trace	As specified in the arswww.ini TraceDir= file
arsMSVR.err	STDERR path in the MidServer start procedure
arsMSVR.out	STDOUT path in the MidServer start procedure
MVS job output	SDSF arsMSVR job

18.4.3 ODWEK trace

Sometimes, it is necessary to collect data for IBM Content Manager OnDemand Web Enablement Kit (ODWEK). Gather this information to assist with the troubleshooting process before you contact IBM Support.

Remember that IBM Support cannot debug custom application code. The purpose of this information is to collect diagnostic test results to help identify a possible problem in ODWEK and provide additional documentation to IBM Support. This information is in IBM Technote 1240220:

<http://www.ibm.com/support/docview.wss?uid=swg21240220>

By using the ODConfig class, you can set the trace as shown in Example 18-4.

Example 18-4 Setting up trace in the ODConfig class

```
ODConfig cfg = new ODConfig(
    /*AfpViewer*/          ODConstant.PLUGIN,
    /*LineViewer*/         ODConstant.APPLET,
    /*MetaViewer default*/ null,
    /*MaxHits*/            500,
    /*AppletDir*/          "/applets",
    /*Language*/           "ENU",
    /*TempDir*/            "c:\\temp",
    /*TraceDir*/           "c:\\path\\to\\trace",
    /*TraceLevel*/          4);
```

Your ODWEK application must be recompiled and restarted for the changes to take effect.

After you enable tracing, re-create the issue and send all arswww.trace* files to IBM Support.

In ODWEK V8.5.0.0 and later, the trace file is written in binary. To convert the trace file from its binary format, use the **arstfmt** command that is in the **bin** directory of your Content Manager OnDemand server (version 8.5 and later). The following sample command shows how to convert the ODWEK trace file:

```
/usr/lpp/ars/v9.5/bin/arstfmt -i arswww.trace -o arswww.trace.txt
```

Tracing can be set to different levels with the **trace** parameter. When you troubleshoot an ODWEK issue, set the trace level to the highest level, unless you are instructed otherwise by IBM Support.

Setting the trace at lower levels, such as **Trace=1**, creates a minimal impact while it alerts you only about error conditions. Setting the lower trace levels are ideal for monitoring an ODWEK application that is in a steady state. Higher levels are used for troubleshooting an ODWEK issue.

18.4.4 TCP/IP packet trace

If a problem exists with TCP/IP, a TCP/IP packet trace might be helpful in troubleshooting.

Example 18-5 show a procedure for tracing to an external writer.

Example 18-5 Sample writer procedure

```
//CTWTR1 PROC
//IEFPROC EXEC PGM=ITTRCWR
//TRCOUT01 DD DSNAME=SYS1.CTRACE1,VOL=SER=xxxxxx,
//           UNIT=xxxxx,SPACE=(CYL,(100),,CONTIG),
//           DISP=(NEW,CATLG)
//SYSPRINT DD SYSOUT=*
```

To obtain a TCP/IP packet trace, complete the following steps:

1. Start your CTRACE external writer procedure by running the following command:

```
TRACE CT,WTRSTART=CTWTR1
```

2. Start and connect a packet component trace to your writer procedure by running the following command:

```
TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIP_Proc)
      reply,WTR=CTWTR1,END
```

3. Start and connect a SYSTCPIP component trace to your writer procedure by running the following command:

```
TRACE CT,ON,COMP=SYSTCPIP,SUB=(TCPIP_Proc)
      reply,WTR=CTWTR1,JOBNAME=(RM_Jobname)
      reply,WTR=CTWTR1,OPTIONS=(socket,pfs,tcp,sockapi),END
```

4. Check whether the component trace is ready to gather data by running the following command:

```
D TRACE,WTR=CTWTR1
```

The display must show a status of ACTIVE.

5. Set packet trace filters by running the following command:

```
V TCPIP,TCPIP_Proc,PKT,clear      # resets filters to none
V TCPIP,TCPIP_Proc,PKT,ON,ip=10.253.0.35 (client IP address)
```

6. Run your recreation scenario.

7. Stop the packet trace by running the following command:

```
V TCPIP,TCPIP_Proc,PKT,clear          # resets filters to none
```
8. Disconnect SYSTCPIP CTRACE from the external writer by running the following command:

```
TRACE CT,ON,COMP=SYSTCPIP,SUB=(TCPIP_Proc)
      reply,WTR=DISCONNECT,JOBNAME=(),OPTIONS=(),END
```
9. Disconnect SYSTCPDA CTRACE from the external writer by running the following command:

```
TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIP_Proc)
      reply,WTR=DISCONNECT,END
```
10. Stop the CTRACE external writer by running the following command:

```
TRACE CT,WTRSTOP=CTWTR1,FLUSH
```

Send the non-formatted packet trace dataset information to IBM Support.

18.4.5 Language Environment (z/OS only)

Many of the Content Manager OnDemand utilities use the Language Environment, and it offers its own customized traces.

To start the trace, rerun the job with the following **DD** statement. Set the **CEE** environmental variable in the JCL (or use **env var** for the UNIX System Services command line):

```
//STDENV DD *
 _CEE_RUNOPTS='HEAPCHK(ON),HEAPPOLLS(OFF)'
```

The output writes to the job log.

This trace is good for problems with starting a service:

```
//CEEDUMP DD
//ARSSOCKD EXEC ...,PARM='TRACE(ON,8M,,LE=1)'/
```

18.4.6 ARSSUPPORT utility

You can use the **ARSSUPPORT** utility to gather log entry diagnostic information. The **ARSSUPPORT** utility is in the **arssupport.jar** file. To run the utility, run the following command:

```
java -jar arssupport.jar
```

The following prerequisites apply to using the command:

- ▶ Ensure that you have Java Runtime Environment version 1.5 or higher to run this program.
- ▶ Ensure that you are logged on to the operating system by using an ID that has administrator authority on Windows or root authority on UNIX.
- ▶ On Windows systems, run the **ARSSUPPORT** utility from the Content Manager OnDemand command prompt.
- ▶ To retrieve system log entries, ensure that the Content Manager OnDemand server is running.
- ▶ The data is collected from the computer where the **ARSSUPPORT** utility is run.

The **ARSSUPPORT** utility generates information about a Content Manager OnDemand server. This information includes information about its configuration and system environment.

ARSSUPPORT archives all files into one compressed file, arssupport.zip and places this file in the odsupport subdirectory of the output directory.

When you get the compressed file, send it to IBM Support.

18.4.7 ARSJESD

The **ARSJESD** program is the server component of Download. Download can be used to transmit output datasets of application programs automatically from the JES spool to Content Manager OnDemand server file systems. If problems occur during the transmission of the **ARSJESD** program, complete the following steps:

1. Stop the **arsjesd** program.
2. Add the **-t** parameter, for example:

```
arsjesd -d /tmp/1 -d /tmp/2 -p 6001 -t
```

3. If you are using Windows, add the **-t** parameter to the following entry in your registry, and then uninstall and reinstall the **arsjesd** service by using the configurator:

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\OnDemand_for  
Windows\@SRV@_ARCHIVE\Services\arsjesd (ARCHIVE)\ProgramParms
```

The example is from an instance named ARCHIVE. Go to the registry key for your respective instance.

4. Start the **arsjesd** service.

The trace file is named `trace.log.<port #>` and is written to the first **arsjesd** directory that is specified by the **-d** parameter. In the example from step 2, this directory is `/tmp/1`.

5. Re-create the issue.

18.4.8 PDF Indexer trace

If a problem occurs during the indexing and loading of PDF documents, you might want to run the PDF Indexer trace.

The PDF Indexer tracing can be performed by using either of two methods:

- ▶ Add the following lines to the indexing parameters:

```
TRACEDD=<trace file name>  
TRACELEVEL=PDF=15
```

For example:

```
TRACEDD=\temp\pdf_tracefile.bin  
TRACELEVEL=PDF=15
```

- ▶ Run the PDF Indexer from the command line and add the trace parameters:

```
arspdoci parmdd=filen.parms inputdd=filen.pdf outputdd=filen.out  
indexdd=filen.ind tracedd=filen.trace tracelevel=pdf=15
```

Where:

- **arspdoci**: Name of the command-line version of the PDF Indexer program
- **parmdd**: Specifies the name of the input file that contains the indexing parameters
- **inputdd**: Specifies the name of the PDF input file to process
- **outputdd**: Specifies the name of the output file that contains the indexed PDF documents that are created by the PDF Indexer

- indexdd: Specifies the name of the output file that contains the index information that is loaded into the database
- tracedd: Specifies the name of the output file that contains the trace information

18.4.9 Trace resolver

The trace resolver output is useful and can be used by IBM Support, programmers, and networking system programmers to diagnose problems in resolving IP host names to IP addresses or IP addresses to IP host names.

The trace resolver helps determine the values of the TCPIP.DATA statements and where the values were obtained.

The details of collecting this trace are in IBM Technote II13398:

<http://www.ibm.com/support/docview.wss?uid=isglII13398>

18.4.10 Conclusion

The traces are enabled for troubleshooting and information gathering. When the task is complete, do not forget to disable the traces.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *Content Manager OnDemand Backup, Recovery, and High Availability*, SG24-6444
- ▶ *Federated Content Management: Accessing Content from Disparate Repositories with IBM Content Federation Services and IBM Content Integrator*, SG24-7742
- ▶ *IBM Content Manager OnDemand Web Enablement Kit Java APIs: The Basics and Beyond*, SG24-7646
- ▶ *IBM System Storage DR550 Setup and Implementation*, SG24-7091
- ▶ *Image and Workflow Library: Content Manager for ImagePlus on OS/390 Implementation and EIP*, SG24-4055
- ▶ *Implementing Content Manager OnDemand Solutions with Case Studies*, SG24-7511
- ▶ *Implementing Web Applications with CM Information Integrator for Content and OnDemand Web Enablement Kit*, SG24-6338
- ▶ *OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization*, SG24-5178

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ *Adobe Press, Adobe Type 1 Font Format*, Addison-Wesley, 1990, ISBN 0201570440
- ▶ *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426
- ▶ *IBM Content Manager OnDemand Messages and Code*, SC19-3356
- ▶ *IBM Content Manager OnDemand - Distribution Facility Installation and Reference*, SC19-3358
- ▶ *IBM Content Manager OnDemand - Web Enablement Kit Implementation Guide*, SC19-3353
- ▶ *IBM Content Manager OnDemand - Windows Client Customization Guide*, SC19-3357
- ▶ *Content Manager OnDemand for i - Planning and Installation Guide*, SC19-2790

- ▶ *Content Manager OnDemand for i - Common Server Administration Guide*, SC19-1292
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Administrative Guide*, SC19-3352
- ▶ *IBM Content Manager OnDemand - Indexing Reference*, SC19-3354
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide*, SC19-3342
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Installation and Configuration Guide for Windows Servers*, GC27-0835
- ▶ *IBM Content Manager OnDemand for Multiplatforms - Introduction and Planning Guide*, SC19-3351
- ▶ *IBM Content Manager OnDemand for z/OS Configuration Guide*, SC19-3363
- ▶ *IBM Content Manager OnDemand for z/OS - Introduction and Planning Guide*, SC19-3365
- ▶ *IBM Content Manager OnDemand - OnDemand Distribution Facility, Installation and Reference*, SC19-3358
- ▶ *IBM Content Manager OnDemand - Web Enablement Kit Implementation*, SC19-3353
- ▶ *IBM DB2 UDB for z/OS and OS/390 - Administration Guide*, SC26-9931
- ▶ *IBM Tivoli Storage Manager for AIX Administrator's Reference*, SC32-0123
- ▶ *Object Access Method Application Programmer's Reference*, SC35-0425-08
- ▶ *OnDemand for z/OS Administration Guide*, SC19-3364
- ▶ *OS/390 OpenEdit Command Reference*, SC28-1982
- ▶ *PDF Reference: Sixth Edition, Adobe Portable Document Format Version 1.7*, November 2006
http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
- ▶ *Tivoli Storage Manager for AIX Administrator's Guide*, GC32-0768
- ▶ *Tivoli Storage Manager for Windows Administrator's Guide*, GC32-0782
- ▶ *Tivoli Storage Manager for Windows Quick Start*, GC32-0784
- ▶ *UNIX System Services Command Reference*, SC28-1892
- ▶ *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- ▶ *z/OS MVS System Commands*, SA22-7627

Online resources

These websites are also relevant as further information sources:

- ▶ DB2 11 for z/OS information
<http://www.ibm.com/software/db2zos/library.html>
- ▶ IBM Content Manager OnDemand production information
<http://www.ibm.com/software/products/us/en/ondemand>
- ▶ Content Manager OnDemand for i Knowledge Center
<http://www.ibm.com/support/knowledgecenter/SSB2EG/welcome>
- ▶ Content Manager OnDemand for Multiplatform Knowledge Center
<http://www.ibm.com/support/knowledgecenter/SSEPCD/welcome>

- ▶ Content Manager OnDemand for z/OS Knowledge Center
<http://www.ibm.com/support/knowledgecenter/SSQHWE/welcome>
- ▶ Ricoh website for Infoprint product information
<http://rpp.ricoh-usa.com/>
- ▶ IBM System i Navigator and IBM Navigator for i information
<http://www.ibm.com/systems/i/software/navigator/>
- ▶ IBM Tivoli Storage Manager home page
<http://www.ibm.com/software/products/en/tivostormana>
- ▶ z/OS information
<http://www.ibm.com/systems/z/os/zos/>
- ▶ *Creating PDF Indexing Parameters Using Floating Triggers*
<http://ibm.co/1FHsXDq>
- ▶ *Understanding the ACIF Input Exit for DB2 Content Manager OnDemand*
<http://ibm.co/1UUcCT0>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

IBM Content Manager OnDemand Guide

SG24-6915-04

ISBN 0738440957



(0.5" spine)
0.475" <-> 0.875"
250 <-> 459 pages



SG24-6915-04

ISBN 0738440957

Printed in U.S.A.

Get connected

